

Less Data Less Tokens: Multilingual Unification Learning for Efficient Test-Time Reasoning in LLMs

Anonymous ACL submission

Abstract

This paper explores the challenges of test-time scaling of large language models (LLMs), regarding both the data and inference efficiency. We highlight the diversity of multi-lingual reasoning based on our pilot studies, and then introduce a novel approach, L^2 multi-lingual unification learning with a decoding intervention strategy for further investigation. The basic idea of L^2 is that the reasoning process varies across different languages, which may be mutually beneficial to enhance both model performance and efficiency. In specific, there are two types of multi-lingual data: the entire long chain-of-thought annotations in different languages and the step-wise mixture of languages. By further tuning based on them, we show that even small amounts of data can significantly improve reasoning capabilities. Our findings suggest that multilingual learning reduces both the required data and the number of inference tokens while maintaining a comparable performance. Furthermore, L^2 is orthogonal to other data efficient methods. Thus, we also emphasize the importance of diverse data selection. The L^2 method offers a promising solution to the challenges of data collection and test-time compute efficiency in LLMs.

1 Introduction

Scaling up training-time and test-time compute are two complementary strategies for enhancing the performance of large language models (LLMs). Training-time scaling allows the model to learn various knowledge through a massive corpus, but it often leads to unsatisfactory reasoning during inference, sometimes causing absurd mistakes. One explanation for this is that conventional inference primarily relies on pattern recognition from memory. In contrast, test-time scaling (e.g., OpenAI o1 (Team, 2025a)) significantly improves reasoning generalization by mirroring human cognitive processes, where problem-solving is not always a direct input-to-output mapping as in supervised

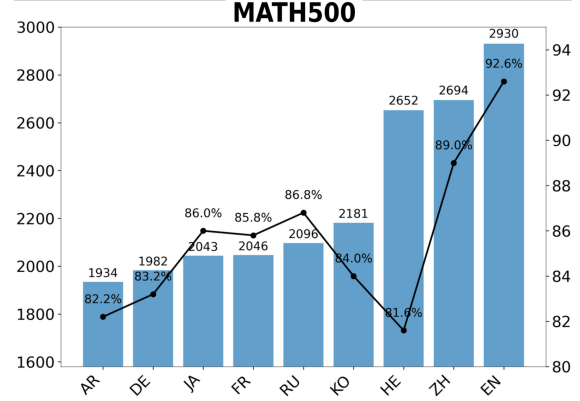


Figure 1: Pilot experimental results of Deepseek-R1-32b on MATH500 dataset using different languages.

fine-tuning, but instead involves iterative reflection and error correction, with a longer thinking process (measured by the number of predicted tokens) guiding the model toward the correct answer.

Much research has explored this idea, revealing two key challenges. The first is the heavy burden of data collection. Some attempts to replicate o1 require up to 747k training samples (Guan et al., 2025), while deepseek R1-32b necessitates 80k samples to achieve o1-level performance (DeepSeek-AI et al., 2025). To reduce the costly long chain-of-thought (CoT) annotations, Sky-T1 (Team, 2025a) distilled 17k samples from QwQ-32b (Team, 2025b) using well-designed data selection strategies. S1 (Muennighoff et al., 2025) further reduced the tuning dataset size to 1,000 by carefully selecting only high-quality, difficult, and diverse samples. Competition continues, with the latest work, LIMO (Ye et al., 2025), demonstrating that as few as 817 samples can enable the model to acquire long reasoning capabilities and tackle highly challenging math problems. As the demand for annotations decreases, an interesting question arises: What is the limit of “less” data?

Another key challenge is the efficiency of test-time compute. As the reasoning chain expands, solving a problem often requires tens of thousands

of tokens, significantly increasing the burden on inference efficiency. For ordinary problems, o1-type models use 1953% more tokens than traditional models to arrive at the same answer (Chen et al., 2025). Higher performance on math competition problems often requires tens of thousands of tokens; thus, reducing inference tokens without sacrificing performance is crucial.

In this paper, we simplify the learning of test-time compute with Less data and Less inference tokens, namely L^2 , through multilingual unification learning. Our core idea is that logical thinking varies across different languages, leading to various solutions and inference token lengths given the same query. As shown in figure 1, our pilot study translates English math questions into other languages, which are prompted to Deepseek-R1-32b to seek solutions in their own languages. We can see the performance and efficiency vary a lot on the AIME24 dataset, ranging from 73.3% accuracy (French) to 40.0% (Hebrew), and from around 7k to 9k inference tokens (Section 2).

Therefore, we assume that augmenting a small amount of CoT data using multiple languages not only enhance data diversity, but also leverage the more concise thinking patterns in certain languages to help inference efficiency.

To test our assumption, we propose a three-step L^2 multilingual unification learning: (1) collecting high-quality English samples (e.g., 6 from OpenAI o1, 1k from s1), (2) generating multilingual CoT annotations using Deepseek API, and (3) creating multilingual data by translating selected reflection steps and tagging them with language tokens; additionally, we introduce a decoding intervention strategy to guide language-specific inference.

We have conducted extensive experiments. Here are our main findings: **1)** Through data augmentation in different languages, only six high-quality samples can improve long reasoning performance by 20%. **2)** Multilingual enhancement is orthogonal to other learning strategies. By introducing more high-quality samples, the performance of our L^2 -32B can be continuously improved, reaching comparable 53% with 651 samples. **3)** While limited data can evoke extended reasoning, performance eventually plateaus; simply increasing samples or languages yields minimal gains, highlighting the need for more diverse data selection or construction. **4)** Multilingual learning enhances performance and notably reduces inference token

usage compared to single-language learning. **5)** Once trained with multi-lingual data, it is unnecessary to infer with different languages. Our major contributions can be summarized as follows:

1. We highlight the differences in reasoning across languages, which not only helps enhance data diversity but also has the potential to improve reasoning efficiency.
2. We propose the namely L^2 paradigm, which is orthogonal to other efficient data methods.
3. We constructed several datasets with different languages and scale. Based on them, we trained models to gain valuable insights for future research.

2 Preliminary Observation on Multi-lingual long Reasoning

We begin by evaluating multi-lingual long CoT reasoning as pilot studies mentioned in the introduction. Specifically, we translate the AIME, GPQA, and MATH500 datasets into nine languages ^[2] and investigate how language choice affects accuracy, normal stopping rates, and token usage in each language. We also compare models of varying scales to examine the influence of multilingual factors on extended reasoning chains.

2.1 Setup

To assess multilingual long-form CoT reasoning, we adopt a selection of open-source models varying in size and pretraining architecture, chosen for their demonstrated reasoning strength and suitability for local evaluation setups:

- **Qwen2.5-based Models** with parameter sizes of 1.5B, 7B, 14B, and 32B, including the Deepseek R1 Distilled Model, which is primarily trained on Chinese and English.
- **LLaMA-based Models** with parameter sizes of 8B and 70B, representing models pretrained on diverse multilingual corpora.

During inference, we record whether the model ends at an appropriate end-of-sequence marker (reporting the proportion of such “normal stops”), and we quantify tokens generated in each language to assess whether reasoning genuinely unfolds in the target language. Due to the space limitation, we only report the results of Deepseek-R1-32B

as representative models due to its strong performance. Other results can be found in Appendix. Note that the scores are based on our careful re-implementation, which may be different from the report due to varied prompts or other config.

2.2 Observation

As shown in Figure 2, we can see that:

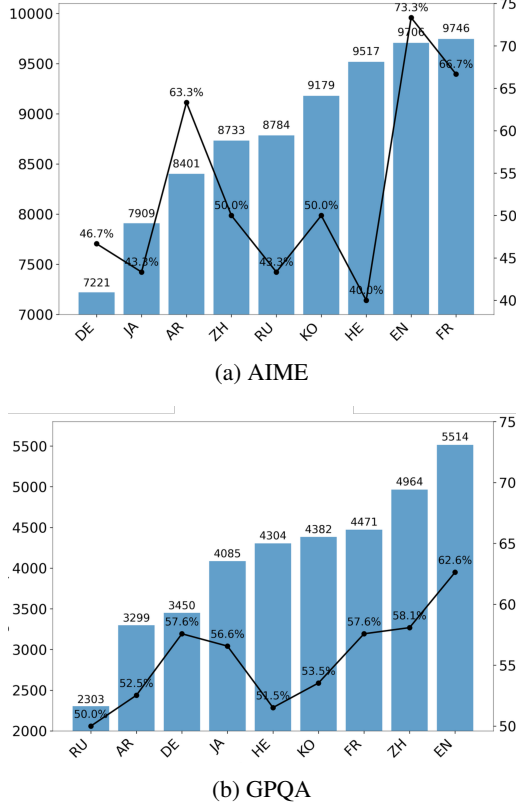


Figure 2: Results of Deepseek-R1-32b on AIME and GPQA datasets using different languages.

Accuracy. Our analysis indicates that English and Chinese achieve superior performance on the GPQA and MATH500, consistent with their dominance in the pre-training corpora. Conversely, the AIME dataset shows notable exceptions: French, Hebrew, and Korean demonstrate unexpectedly competitive accuracies. We attribute these deviations primarily to AIME’s limited size of only 30 problems, which may increase statistical variance and impact the stability of accuracy estimates.

Normal Stopping and Token Usage. Most outputs terminate correctly (though sometimes excessively or repetitively), but token usage varies notably across languages.

Multilingual Reasoning and Code-Switching. For Chinese, English, Korean prompts, the model predominantly reasons in that language; however,

for other languages, sometimes the LLMs reverts to Chinese or English mid-way, sometimes mixing languages in a single CoT.

In conclusion, results indicate significant variations in accuracy and inference length across languages, suggesting distinct advantages. However, LLMs’ occasional confusion between languages presents challenges for controlled multilingual reasoning, which will be discussed later.

3 Data and Method

To combine the merits of reasoning in different languages, our proposed L^2 multi-lingual unification learning is first to augment long CoT data at both the entire solution level and at the step level, then finetune LLMs using the augmented data. The overall framework consists of three key steps: high-quality sample collection, multi-lingual thoughts annotation, and multi-lingual unification learning. Next we will introduce them in turn, followed by multi-lingual decoding interventions to explore the impacts of languages on inference.

3.1 High-quality sample collection

We collect three sets of data with different scales from existing resources. Note that we didn’t combine them together into a single set. Instead, we investigate our method using them separately to verify the effectiveness.

- $L^2 - Mo1_6^l$. This set contains six official examples adapted from OpenAI’s website, manually curated and formatted in $\LaTeX$. The topics include *Cipher*, *Coding*, *Math*, *Crossword*, *English*, and *Science*, with one question per topic. The superscript l denote the number of languages in experiments using the following two-step augmentation. For instance, $L^2 - Mo1_6^4$ involves four language (ZH, EN, KO, RU), results in 2,700 multi-lingual samples in total.
- $L^2 - MS1_{samples}^l$. We introduce 100, 651, and 1000 samples from the “S1k” dataset (Muenighoff et al., 2025), focusing primarily on mathematical problems, to evaluate how the number of samples affects model training effectiveness. Initially, we included only partial data due to instability issues with the Deepseek API used for generating Chain-of-Thought (CoT) reasoning paths, resulting in only 651 valid instances. Subsequently, after the API’s stability was restored, CoT paths were generated for the full set of 1k

samples. In experiments, such as $L^2-MS1_{651}^4$, we introduce four languages: English, Chinese, Russian, and Korean. We did not select all nine languages mainly due to considerations regarding computational cost and efficiency. Additionally, we aimed to balance the sizes of these two training sets for comparative purposes.

- $L^2 - MBS_{500}^l$. We randomly select 500 questions from Bespoke-Stratos-17k (Labs, 2025) as the data set. all other configurations remain consistent with $L^2-MS1_{samples}^l$.

3.2 Multi-lingual thoughts annotation

We curate multilingual CoT at the solution level by translating questions with GPT-4o, generating step-by-step explanations via Deepseek API in target languages, and collecting diverse reasoning paths without rigorously evaluating translation quality.

3.3 Multi-lingual unification learning

We curate multilingual unification data by segmenting English CoT texts into reflection fragments, randomly translating selected steps (identified by cues like “Wait,” “Hmm”) via GPT-4o, and marking language boundaries with special tokens, thereby creating a code-switched corpus to foster flexible cross-lingual reasoning (illustrated in Figure 3).

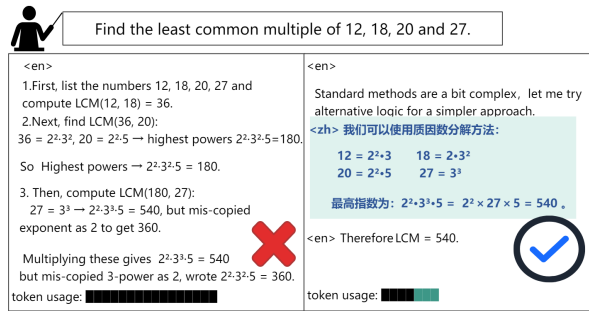


Figure 3: Comparison of reasoning strategies: mixed Chinese-English reasoning (right) achieves correct results with clearer logic and fewer tokens than English-only (left).

Training After the above two steps, we will obtain the entire CoT in English and Chinese, respectively, as well as the step-wise mixture of thoughts in two languages. Here, we introduce the training details. We utilize the llamafactory framework, integrating flash attention and a light kernel acceleration package to expedite training. Our approach follows standard Supervised Fine-Tuning (SFT) with ZeRO Stage 3 optimization, and we set the maximum sequence length to 16k tokens. Training is conducted on 8 H20 GPUs.

For datasets with fewer than 300 training samples (*small datasets*), we set batch size and gradient accumulation step to 1, over-sample data to ensure sufficient coverage, and train until loss approaches zero. For larger datasets, we keep batch size at 1 but increase gradient accumulation step to 12 and train for 3 epochs.

3.4 Decoding Intervention

Algorithm 1 Decoding Intervention Pseudocode

```

Require: logits,  $\alpha$ ,  $\beta$ ,  $k$ , zh_token
Ensure: adjusted_logits
1: top_tokens  $\leftarrow$  TopK(logits,  $k$ )
2: if zh_token  $\in$  top_tokens then
3:    $u \leftarrow$  Uniform(0, 1)
4:   if  $u < \alpha$  then
5:     logits[zh_token]  $\leftarrow$  logits[zh_token] +  $\beta$ 
6:   end if
7: end if
8: return logits

```

Figure 4: Comparison of reasoning strategies: mixed Chinese-English reasoning (right) achieves correct results with clearer logic and fewer tokens than English-only (left).

We propose a *decoding intervention* during inference that adjusts language switching probabilities using special language tokens and hyperparameters. Specifically, given $\alpha \in [0, 1]$ controlling boost or suppression likelihood, magnitude β for logit adjustment, and a top- k cutoff, we sample $u \sim \text{Uniform}(0, 1)$ whenever a language token is within the top- k candidates. If $u < \alpha$, we boost the token’s logit by $+\beta$; otherwise, we penalize it by $-\beta$, thus shaping language usage.

4 Experiments

This section details the experimental setup, baseline methods (§4.1), and key results, with a particular focus on the performance under varying number of languages and data sizes.

4.1 Baselines

To assess the effectiveness of our low-data multilingual long-chain-of-thought approach, we compare against several representative baselines:

- **OpenAI-o1** (OpenAI, 2024): A closed-source commercial large language model, the first to provide long-chain reasoning services.
- **Open-source model**: The base model Qwen2.5-32B (Qwen et al., 2025), the QWQ model with the same 32B size, and the powerful O1-level open-source model, Deepseek R1 (DeepSeek-AI et al., 2025).

- **Data-efficient models:** Sky-T1, s1, and LIMO, which were fine-tuned with as little as 17k, 1k, or even fewer data (Muennighoff et al., 2025; Team, 2025a; Ye et al., 2025), achieving performance comparable to o1-level models.

4.2 Setup

We largely follow the experimental setup of s1 (Muennighoff et al., 2025) for fair comparison. We choose Qwen2.5-32B as our base model and finetuned using $L^2-Mo1_6^{10}$ and $L^2-MS1_{651}^4$ (introduced in Section 3.1, respectively, resulting in three well-trained models, $L^2-32B-Mo1_6^{10}$, $L^2-32B-MS1_{651}^4$ and $L^2-32B-MS1_{1k}^4$. For assessment, we use the standard framework vllm for inference with a temperature of 0.7, recording only the model’s first response. Our evaluation covers four datasets — AIME24 (30), GPQA DIAMOND (198), and MATH500 (500).

We evaluate AIME and GPQA via string parsing, manually check decimals for MATH500, and use annotators for Graduate Entrance Exam tasks.

4.3 Main Results

Table 1 shows the overall results. We can see that: **1)** with only 6 samples (although augmented to 2,700 samples), our model $L^2-32B-Mo1_6^{10}$ greatly improves the performance over the base model by 16.6%, 18.2%, and 12%, respectively. **2)** By introducing more high-quality data (i.e., 612 samples augmented to 4,500), we achieve comparable performance with models using much more data. This demonstrates the effectiveness of our multi-lingual unification learning. **3)** The strongest models are still those using much more data, like r1 or o1. Combined with the above conclusion, this suggests the importance of both curation of diverse data and how to select the high-quality ones.

4.4 RQ1: How does extremely small training data affect test-time scaling?

In this experiment, we focus on the $L^2-Mo1_6^{10}$ dataset. Qwen2.5-32b is our base model. To ensure fair comparison, we finetune it using the six samples with upsampling, resulting in Qwen2.5-32b-o1₆.

³Based on manual inspection, some Math500 standard answers were incorrectly formatted, corresponded to multi-part fill-in answers, or involved decimals with inconsistent precision requirements. As a result, the format-based validator mistakenly flagged originally correct answers as wrong—affecting a non-negligible number of problems (8–12 out of 500; see the appendix for specific cases). The reported results have been corrected accordingly.

Model	# ex.	AIME2024	MATH500	GPQA
API only				
o1-preview	N.A.	44.6	85.5	73.3
o1-mini	N.A.	70.0	90.0	77.0
o1	N.A.	74.4	94.8	77.3
Open Weights				
Qwen2.5-32b*	N.A.	26.7	84.0	49.0
Qwen2.5-32b [#]	N.A.	16.7	76.2	45.5
Qwen2.5-32b	N.A.	10.0	69.0	41.0
QwQ-32B	N.A.	50.0	90.6	65.2
r1	N.A.	79.0	97.3	71.5
r1-distill	~800K	72.0	94.3	62.1
Open Weights and Open Data				
Sky-T1	17K	43.0	82.4	56.8
Bespoke-32B	17K	63.0	93.0	58.1
s1 w/o BF	1K	50.0	92.6	56.6
s1-32B	1K	56.0	93.0	59.6
LIMO	1K	57.1	94.8	66.7
$L^2-32B-Mo1_6^{10}$	6	23.3	87.4	49.5
$L^2-32B-MS1_{651}^4$	651	63.3	93.0	60.0
$L^2-32B-MS1_{1k}^4$	1k	63.3	95.0 ¹ /93.0	61.0

Table 1: Overall performance of our models and base-lines on the AIME 2024, MATH 500, and GPQA Diamond datasets. Note that the three scores of Qwen2.5-32b are due to different implementation. Ours is without any superscript, * denotes the scores in S1 original paper, and [#] denotes the scores from Sky-T1.

Furthermore, $L^2-32b-Mo1_6^1$ represents adding only English CoT data obtained from DeepSeek R1, while $L^2-32b-Mo1_6^4$ incorporates multi-lingual CoT data. As shown in Table 2, we can conclude that:

1) By tuning using six high-quality samples, even with some upsampling techniques, the model Qwen2.5-32b-o1₆ only achieves slight improvements. Compared with our approach augmented with multi-lingual data, $L^2-32b-Mo1_6^4$ achieves significant performance gains across all datasets. This demonstrates the effectiveness of the multi-lingual assumption in improving performance through increased data diversity.

2) Compared with $L^2-32b-Mo1_6^4$, the performance improvement of $L^2-32b-Mo1_6^1$ is much smaller. This indicates that even for the same questions, obtaining diverse reasoning data in multiple languages is crucial to enhance model performance.

Setting	AIME	GPQA	MATH500
Qwen2.5-32b	0.10	0.41	0.69
Qwen2.5-32b-o1 ₆	0.17	0.43	0.74
$L^2-32b-Mo1_6^1$	0.33	0.34	0.67
$L^2-32b-Mo1_6^4$	0.33	0.49	0.85
$L^2-32b-Mo1_6^9$	0.23	0.49	0.87

Table 2: accuracy results when scaling to a total of 6 questions based on the L^2-Mo1_6 dataset, and using multilingual augmented data.

4.5 RQ2: Where is the upper boundary of multi-lingual extension?

4.5.1 Analyzing the Impact of Data Scale

To investigate the impact of data scale on model performance, we randomly selected 100 questions from the S1 dataset as the initial query pool and constructed 10 incremental training datasets. For instance, the dataset labeled as $L^2-MS1_{10}^9$ comprises 10 queries annotated with CoT reasoning in 9 different languages, as described in Section 3.1, using the MCOT method. Similarly, $L^2-MS1_{20}^9$ was created by adding another 10 randomly selected queries from $L^2-MS1_{100}^l$, ensuring no overlap with the previous 10 queries of $L^2-MS1_{10}^9$. This process was iteratively continued, expanding the dataset to include up to 100 queries and resulting in 10 datasets of increasing size. Each dataset was subsequently finetuned and evaluated under consistent experimental settings to ensure fair comparison.

The results demonstrate that around the scale of 30 queries, the model exhibits a distinct inflection point, where both its capabilities and token consumption increase significantly. This phenomenon was consistently observed across various evaluation datasets, including MATH500 (+45.8%), GPQA (+67.8%), AIME24 (+75.0%), and AIME25 (+175.0%) (Figure 5, Appendix). These findings suggest that a modest expansion of high-quality annotated data, particularly beyond the 30-query threshold, substantially enhances model performance by alleviating early-stage data scarcity and enabling the model to better generalize and leverage its reasoning capabilities.

4.5.2 Evaluating Cross-Language Family Effects

We further investigated whether multilingual training across diverse language families improves model performance compared to training within a single language family.

We conducted the following experiment: The nine languages were grouped into three language families. As demonstrated in Section 4.5.1, training with 100 queries enables the model to develop long reasoning chains and improves performance across various datasets. For this experiment, we used the $L^2-MS1_{100}^l$ dataset, which includes all nine languages.

²Nine languages: Chinese (zh), English (en), French (fr), German (de), Arabic (ar), Hebrew (he), Japanese (ja), Korean (ko), Russian (ru).

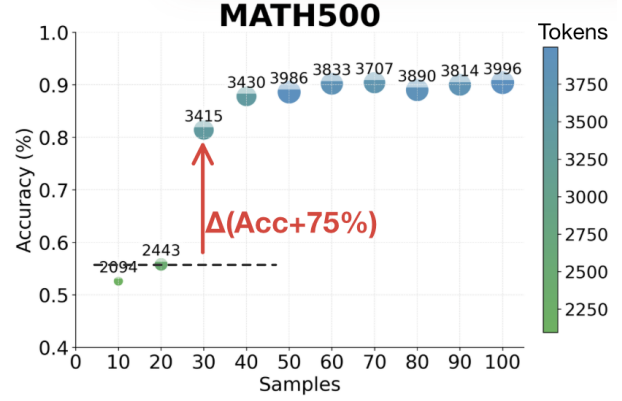


Figure 5: The x-axis indicates the number of questions included in the model training, and the y-axis denotes the achieved accuracy. Point size, shading intensity, and numeric annotations represent the quantity of generated tokens.

- **East-Asian:** Simplified Chinese (zh), Japanese (ja), Korean (ko)
- **Indo-European:** English (en), French (fr), German (de), Russian (ru)
- **Afro-Asiatic:** Arabic (ar), Hebrew (he)

In Figure 6, we generated training datasets by randomly combining different languages across these families and trained a model on each dataset. In the resulting visualization, each shape represents models trained with languages from specific language families. The more language families trained, the higher the accuracy and the fewer tokens used, yielding better results. Models positioned closer to the top-left corner indicate superior performance. Detailed numerical results can be found in the appendix.

4.6 RQ3: Does our strategy orthogonal to other data curation methods?

Existing methods employ different strategies to select high-quality mathematical data. To further validate our approach, we increase the number of initial samples by randomly selecting samples from two typical sources: s1k and Bespoke-Stratos-17k (Labs, 2025; Team, 2025a). We have introduced the augmented s1k dataset $L^2-MS1_{651}^4$ in Section 3.1. For another source, we randomly select 500 samples from the Bespoke-Stratos-17k, marked as BS_{500} , primarily featuring mathematics and programming problems. After multi-lingual augmentation, there are totally 500 samples in $L^2-MBS_{500}^4$. By removing the step of multi-lingual unification, the model’s performance drops significantly. This suggests that the step-wise mixture of

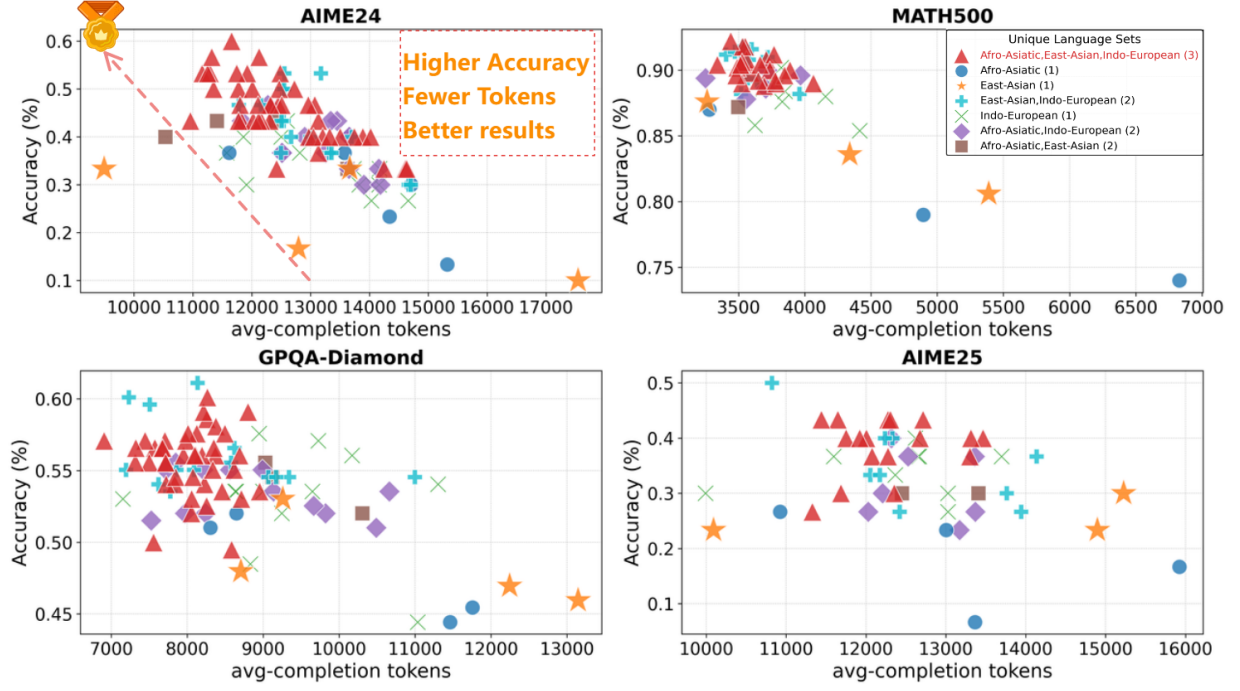


Figure 6: "East-Asian, Indo-European(2)" indicates a dataset including languages from both the East Asian and Indo-European families. Multiple shapes of the same kind indicate the same number of language families but with different combinations of specific languages.

languages contributes to enhancing generalization and reasoning capabilities.

We can see that regardless of the data source, our methods can effectively boost performance. However, it is also noticeable that as the amount of original data increases, the marginal benefit of multilingual learning diminishes. This could be attributed to the model approaching its inherent capacity limits as the training data scale becomes larger.

Setting	AIME	GPQA	MATH500
<i>BS₅₀₀</i> Data Set			
<i>Qwen2.5-32b-BS₅₀₀</i>	0.43	0.52	0.90
<i>L²-MBS₅₀₀⁴-uni</i>	0.46	0.55	0.91
<i>L²-MBS₅₀₀⁴</i>	0.60	0.51	0.91
<i>S1</i> Data Set			
<i>Qwen2.5-32b-S1₁₀₀</i>	0.43	0.54	0.85
<i>L²-32b-MS1₁₀₀⁴</i>	0.53	0.53	0.90
<i>Qwen2.5-32b-S1₆₅₁</i>	0.63	0.56	0.93
<i>L²-32b-MS1₆₅₁⁴</i>	0.63	0.60	0.93
<i>Qwen2.5-32b-S1_{1k}</i>	0.60	0.60	0.91
<i>L²-32b-MS1_{1k}⁴</i>	0.63	0.61	0.93

Table 3: Accuracy results when scaling to a total of 500+6 multi-lingual unification samples from the Bespoke-Stratos-17k resource. Accuracy is evaluated on our dataset.

4.7 RQ4: Can our strategy also benefit inference efficiency?

We hypothesize that long COT annotations from diverse language families offer complementary reasoning patterns, enhancing accuracy and inference efficiency through reduced token usage, unlike augmentations from linguistically similar sources, as shown in Figure 6 (details see appendix).

4.8 RQ5: What if we intervene the decoding by controlling the reasoning languages?

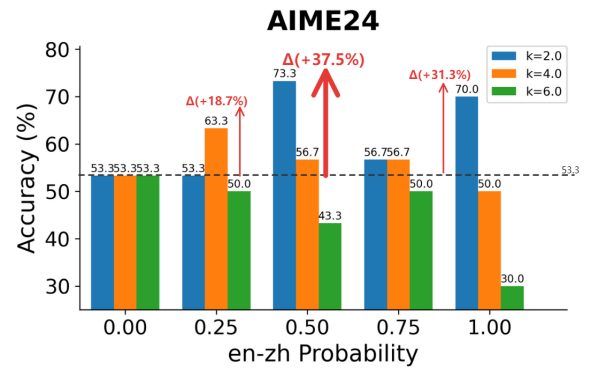


Figure 7: Decoding control over Chinese and English reasoning paths, with evaluation results on the AIME24 dataset.

In Figure 7, We investigate how to guide a model's reflective reasoning to be expressed in a

specific language during decoding. In this setting, the maximum number of generated tokens significantly increases from 15k in previous experiments to 131k. We introduce an intervention ratio α that adjusts the frequency at which tokens prompting the target language appear. During training, we ensure this language is used to encode reflective reasoning. Notably, this approach does not diminish the model’s ability to reason reflectively; it can still generate fluent English. We vary the parameter $k \in \{2, 4, 6\}$ to examine its impact on the model’s behavior. Our findings show that a higher intervention ratio makes it more likely for the model to shift its reasoning into another language (zh). When $k = 2$ or $k = 4$, the model can effectively switch between multiple languages, reaching an accuracy of **73.3%** on the AIME24 dataset (see appendix for case studies on difficult problems with successful solutions). However, at $k = 6$, the reflection tokens (originally assigned a low probability) are activated more frequently, producing extensive reflective segments that interfere with the model’s normal reasoning process.

5 Case study

In Figure 8, The examples show a mathematical problem, where the model answers through multi-language reasoning. The model effectively handles this by utilizing its multi-language reasoning capabilities (detailed case studies provided in the appendix). This approach allows the model to seamlessly process and analyze the mathematical problem across different languages, ensuring accurate and efficient solutions regardless of the language input. By leveraging the strengths of multi-language understanding, the model delivers robust and reliable responses in various linguistic contexts.

A regular hexagon can be divided into six equilateral triangles. If the perimeter of one of the triangles is 21 inches, what is the perimeter, in inches, of the regular hexagon?

<en>Okay, let's see. The problem says a regular hexagon can be divided into six equilateral triangles.....Therefore, the perimeter of the hexagon is 42 inches. <zh>让我再核实一下。如果三角形的周长是 21 英寸，那么每条边就是 7 英寸。由于正六边形的每条边与这个三角形的边相等，因此六边形每条边的长度是 7 英寸。接着，六条边乘以 7 英寸得到 42 英寸。这看起来很简单。所以答案是 42 英寸。<en> so perimeter is 6*7=42. Yep, that's correct. Alternatively, maybe there's a different interpretation. <zh>因为每个三角形都是等边三角形，所以所有三条边的长度都应该相等。因此，这里的三条边应该都相等，也就是说，六边形的边长和半径的长度相等。所以，周长是三条边相加，即 3s = 21。因此，s = 7。由此可得，六边形的周长是 6s = 42。所以，答案仍然是 42。Yes, that's the same result. So regardless of how I think about it, the answer is 42 inches. **Final Answer** The perimeter of the regular hexagon is \boxed{42} inches.

Figure 8: Mathematical problem example

6 Related Work

6.1 Test-time scaling

Test-Time Scaling (TTS) enhances LLM performance by allocating extra computational resources during inference(Liu et al., 2025; Zhang et al., 2025; Wu et al., 2025; Ji et al., 2025). Unlike traditional scaling methods, TTS enables fixed-parameter models to achieve superior outcomes through extended inference-time processing, akin to "thinking longer." (Faria and Smith, 2025; Kim et al., 2024). Diverse TTS strategies include computational budget control, sampling and search methods(Muennighoff et al., 2025; Aggarwal and Welleck, 2025; Son et al., 2025), verification-guided approaches(Wang et al., 2025; Lifshitz et al., 2025), and latent reasoning paradigms. Empirical results show significant reasoning gains, with smaller compute-optimal TTS models surpassing larger models.

6.2 Multilinguality and Logical Reasoning

Recent advancements in large language models show that multilingual strategies significantly enhance logical reasoning(Ghosh et al., 2025; Tran et al., 2025). While these models excel in high-resource languages like English, performance gaps persist for lower-resource languages(Ravisankar et al., 2025). Techniques like cross-lingual thought prompting (XLT)(Huang et al., 2023) and English-pivoted CoT training exploit strong English reasoning to boost multilingual outcomes. Methods such as LayAlign(Ruan et al., 2025) and Ada-CoT(Huang et al., 2025) further align abstract reasoning patterns across languages, promoting culturally responsive and globally applicable models.

7 Conclusion and Future Work

In this paper, we present the L^2 approach, which leverages multilingual unification learning to enhance the test-time scaling of LLMs. Our method is demonstrated in incorporating a minimal amount of data and reducing the number of inference tokens, while maintaining long CoT reasoning capabilities. Our experimental results demonstrate that multilingual data can significantly improve long-reasoning tasks, with only a small number of high-quality samples yielding notable gains in performance. Furthermore, the L^2 approach offers a scalable and efficient path forward for training models that are capable of handling complex tasks while minimizing computational costs.

Limitations

The L^2 approach offers promising efficiency for LLM test-time scaling but faces limitations, including varying language proficiency in base models and differences in tokenization due to linguistic variations, potentially affecting efficiency and results. Despite these, extensive experiments support our hypothesis. Integrating models trained on diverse languages also poses safety and quality risks, especially for low-resource languages, potentially causing biases and errors.

References

Pranjal Aggarwal and Sean Welleck. 2025. [L1: Controlling how long a reasoning model thinks with reinforcement learning](#).

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. [Do not think that much for 2+3=? on the overthinking of o1-like llms](#).

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jia Shi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shutong Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang,

Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#).

Gonalo Faria and Noah A. Smith. 2025. [Sample, don’t search: Rethinking test-time alignment for language models](#).

Akash Ghosh, Debayan Datta, Sriparna Saha, and Chirag Agarwal. 2025. [The multilingual mind : A survey of multilingual reasoning in language models](#).

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. [rstar-math: Small llms can master math reasoning with self-evolved deep thinking](#).

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. [Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting](#).

Xin Huang, Tarun Kumar Vangani, Zhengyuan Liu, Bowei Zou, and Ai Ti Aw. 2025. [Adacot: Rethinking cross-lingual factual reasoning through adaptive chain-of-thought](#). *arXiv preprint arXiv:2501.16154*.

Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Kai Yao, Jia Xu, Linjian Mo, and Min Zhang. 2025. [Test-time compute: from system-1 thinking to system-2 thinking](#).

Eungyeup Kim, Mingjie Sun, Christina Baek, Aditi Raghunathan, and J. Zico Kolter. 2024. [Test-time adaptation induces stronger accuracy and agreement-on-the-line](#).

Bespoke Labs. 2025. [Bespoke-stratos: The unreasonable effectiveness of reasoning distillation](#). <https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation>. Accessed: 2025-01-22.

Shalev Lifshitz, Sheila A. McIlraith, and Yilun Du. 2025. [Multi-agent verification: Scaling test-time compute with multiple verifiers](#).

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025. [Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling](#).

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. [s1: Simple test-time scaling](#).

OpenAI. 2024. Learning to reason with llms. <https://openai.com/index/learning-to-reason-with-llms/>. Accessed: 2024-09-10.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. [Qwen2.5 technical report](#).

Kartik Ravisankar, Hyojung Han, and Marine Carpuat. 2025. [Can you map it to english? the role of cross-lingual alignment in multilingual performance of llms](#).

Zhiwen Ruan, Yixia Li, He Zhu, Longyue Wang, Weihua Luo, Kaifu Zhang, Yun Chen, and Guanhua Chen. 2025. Layaalign: Enhancing multilingual reasoning in large language models via layer-wise adaptive fusion and alignment strategy. *arXiv preprint arXiv:2502.11405*.

Guijin Son, Jiwoo Hong, Hyunwoo Ko, and James Thorne. 2025. [Linguistic generalizability of test-time scaling in mathematical reasoning](#).

NovaSky Team. 2025a. Sky-t1: Train your own o1 preview model within 450. <https://novasky-ai.github.io/posts/sky-t1>. Accessed : 2025-01-09.

Qwen Team. 2025b. [Qwq-32b: Embracing the power of reinforcement learning](#).

Khanh-Tung Tran, Barry O’Sullivan, and Hoang D. Nguyen. 2025. [Scaling test-time compute for low-resource languages: Multilingual reasoning in llms](#).

Yiming Wang, Pei Zhang, Siyuan Huang, Baosong Yang, Zhuosheng Zhang, Fei Huang, and Rui Wang. 2025. [Sampling-efficient test-time scaling: Self-estimating the best-of-n sampling in early decoding](#).

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. [Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models](#).

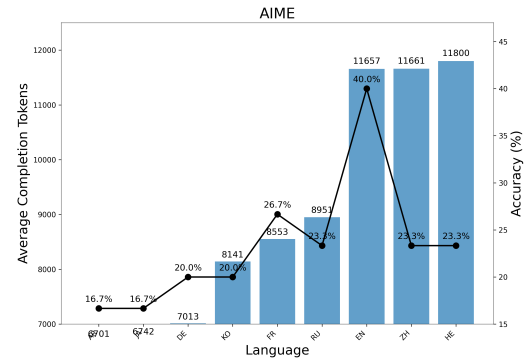
Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. [Limo: Less is more for reasoning](#).

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. 2025. [What, how, where, and how well? a survey on test-time scaling in large language models](#).

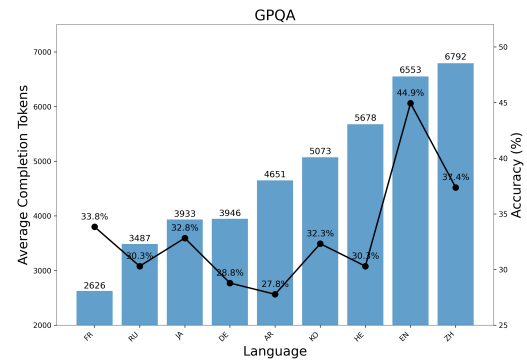
A Appendix

A.1 Accuracy and Token Consumption across Different Models and Languages

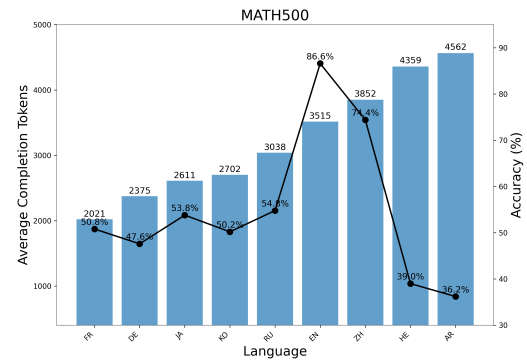
Figures 9, 10, and 11 present detailed results illustrating the accuracy and token consumption of five language models—R1-Llama (8B, 70B) and R1-Qwen (1.5B, 7B, 14B)—evaluated across three benchmarks: AIME, GPQA, and MATH500.



(a) R1-qwen-1.5b-AIME

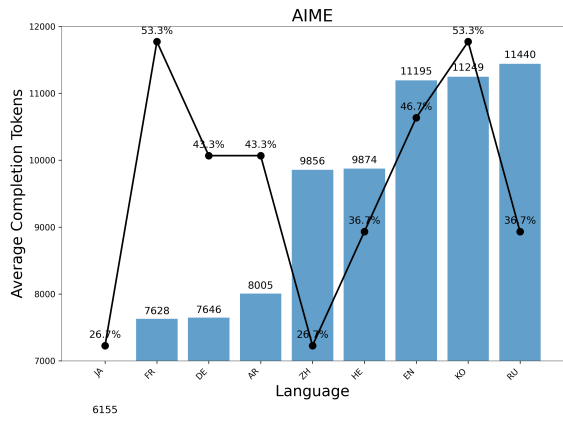


(b) R1-qwen-1.5b-GPQA

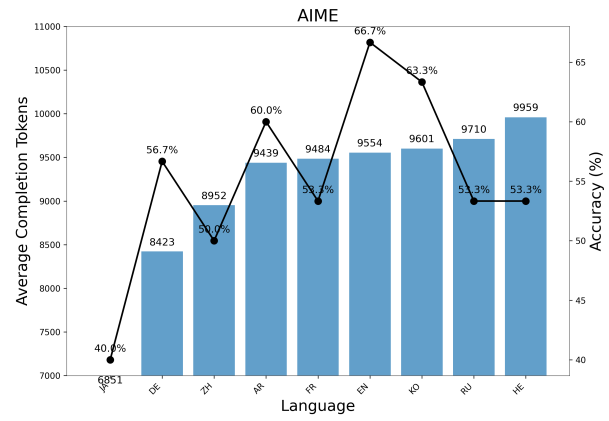


(c) R1-qwen-1.5b-MATH500

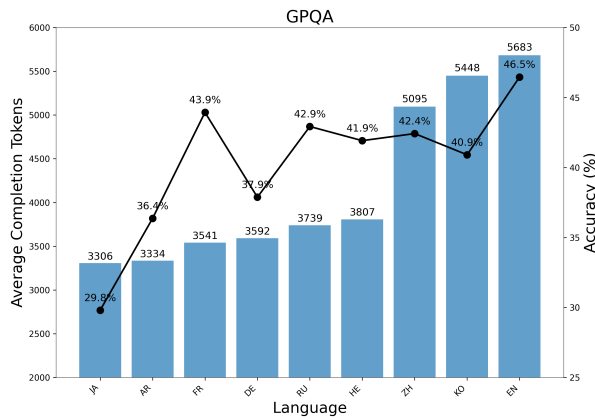
Figure 9: Results of R1-qwen-1.5b on AIME, GPQA, and MATH500 datasets using different languages.



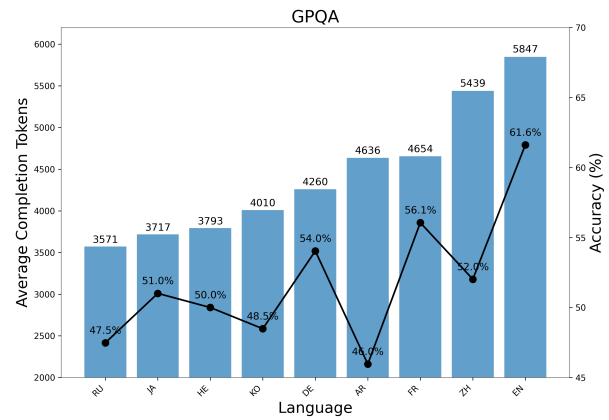
(a) R1-qwen-7b-AIME



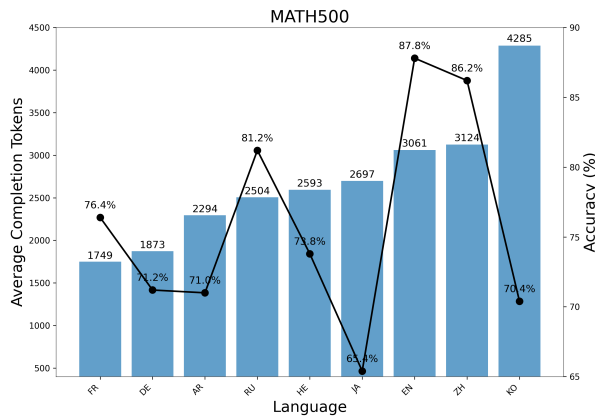
(b) R1-qwen-7b-AIME



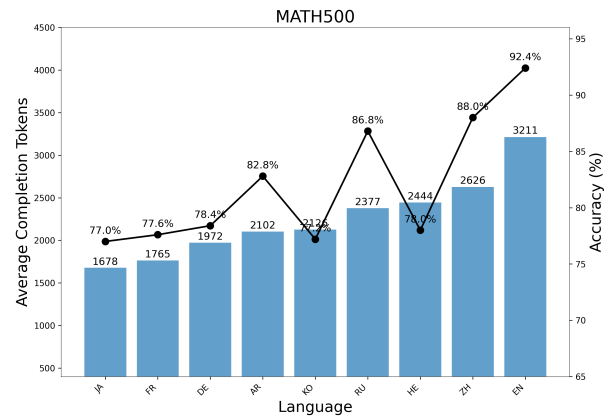
(c) R1-qwen-7b-GPQA



(d) R1-qwen-14b-GPQA



(e) R1-qwen-7b-MATH500



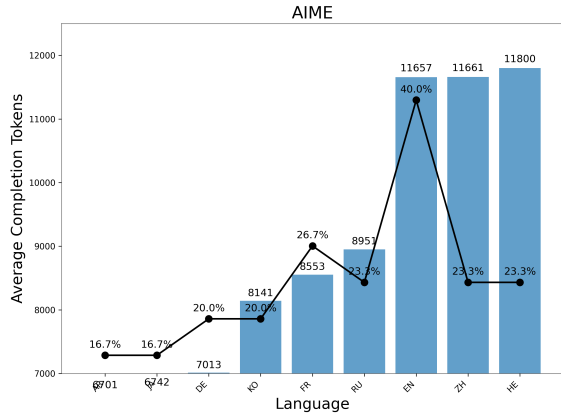
(f) R1-qwen-14b-MATH500

Figure 10: Results of R1-qwen-7b and R1-qwen-14b on AIME, GPQA, and MATH500 datasets using different languages.

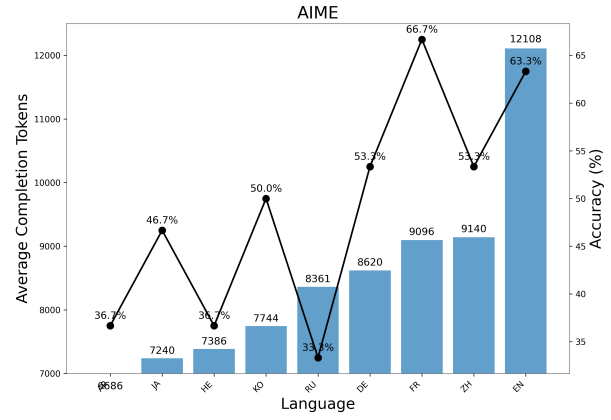
A.2 Accuracy and Token Consumption across Different Models and Languages

around 30 samples.

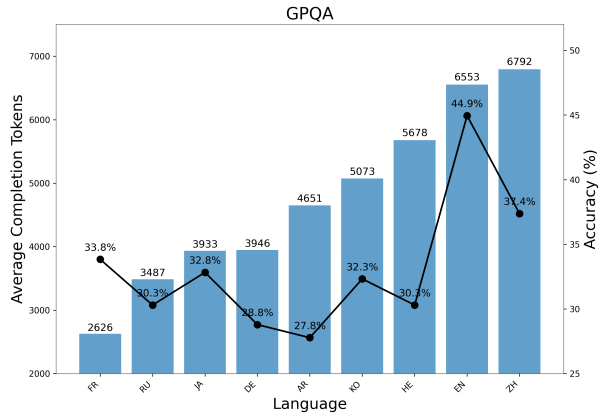
Figure 12 comprehensively illustrates the relationship between the number of training samples, model accuracy, and generated tokens across the AIME24, AIME25, GPQD, and MATH500 benchmarks. Notably, there is a clear inflection point



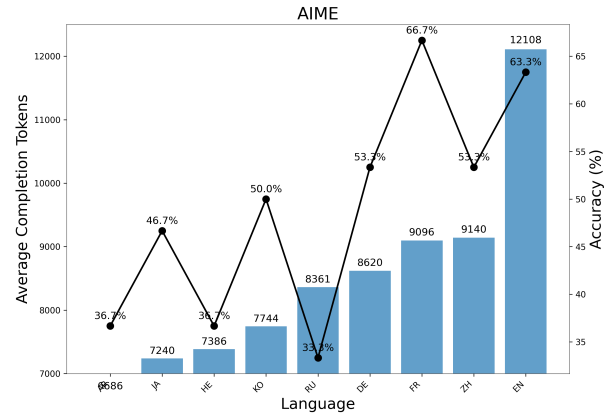
(a) R1-llama-8b-AIME



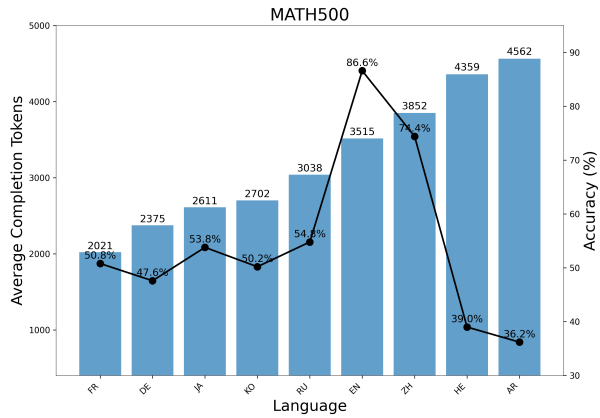
(b) R1-llama-70b-AIME



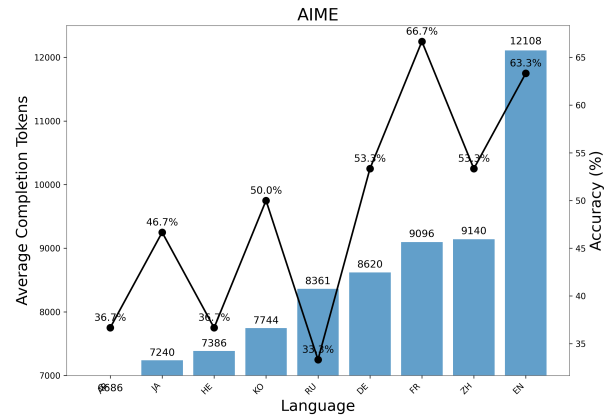
(c) R1-llama-8b-GPQA



(d) R1-llama-70b-AIME



(e) R1-llama-8b-MATH500



(f) R1-llama-70b-AIME

Figure 11: Results of R1-llama-8b and R1-llama-70b on AIME, GPQA, and MATH500 datasets using different languages.

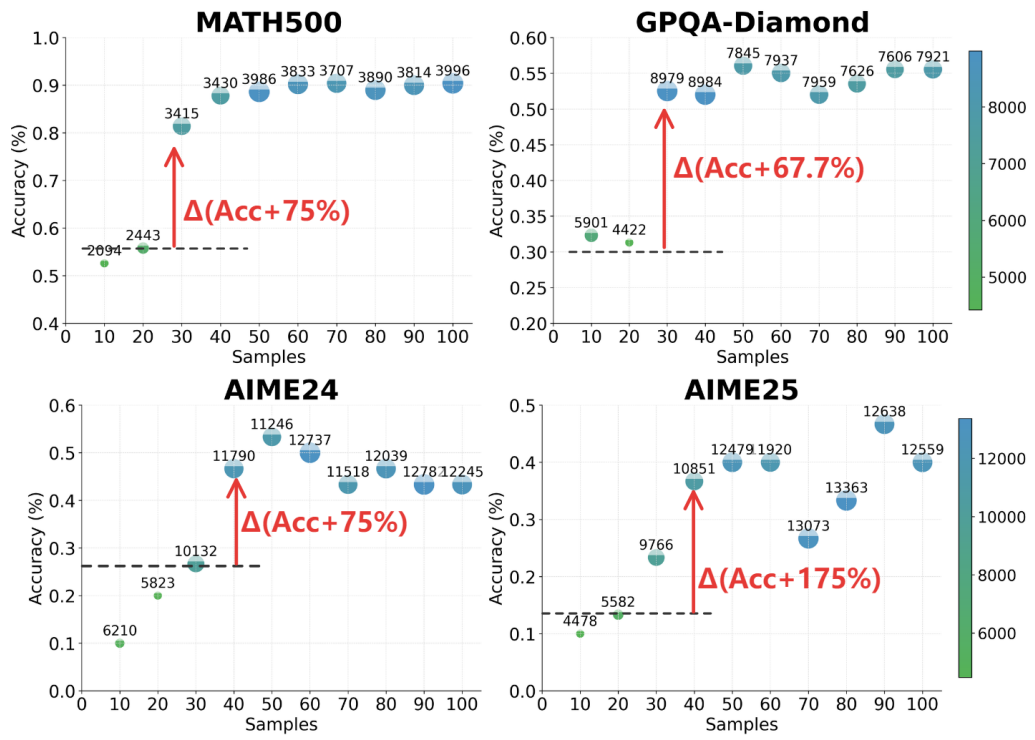


Figure 12: The x-axis indicates the number of questions included in the model training, and the y-axis denotes the achieved accuracy. Point size, shading intensity, and numeric annotations represent the quantity of generated tokens.

Appendix: AIME CASE

738

Source: 2024_AIME_II_Problems/Problem_14

Let $b \geq 2$ be an integer. Call a positive integer n *b-beautiful* if it has exactly two digits when expressed in base b and these two digits sum to $\sqrt{n}$. For example, 81 is 13-beautiful because $81 = \underline{6} \cdot \underline{3} \cdot 13$ and $6 + 3 = \sqrt{81}$. Find the least integer $b \geq 2$ for which there are more than ten *b-beautiful* integers.

<|begin_of_thought|>

Okay, so I need to find the smallest base b (where b is at least 2) such that there are more than ten b -beautiful numbers. A number n is b -beautiful if it has exactly two digits in base b , and the sum of these two digits equals the square root of n . The example given is $n = 81$ being 13-beautiful because in base 13, it's written as 63, and $6 + 3$ equals the square root of 81, which is 9.

嗯，我们先来分析这个问题。既然 n 在基底 b 下是一个两位数，那么它必须满足 b 以内的最大值，也就是 $b^2 - 1$ 。等等，但在基底 b 中，一个两位数可以表示为 $(x \cdot b + y)$ ，其中 (x) 和 (y) 是在 (0) 到 $(b - 1)$ 范围内的数字，而 (x) 不能为 (0) 。因此， n 的取值范围是从 $(b \cdot 1 + 0 = b)$ 到 $((b - 1) \cdot b + (b - 1) = b^2 - 1)$ 。不过实际上，由于 (x) 和 (y) 都不能超过 $(b - 1)$ ，因此 n 的最大值确实是 $(b^2 - 1)$ 。

But n has to be a perfect square because $\sqrt{n}$ must be an integer (since the digits sum to $\sqrt{n}$, and digits are integers). So n must be a perfect square, and in base b , it's a two-digit number. Therefore, the two-digit perfect squares in base b correspond to squares between b and $b^2 - 1$. Wait, but $\sqrt{n}$ would be between $\text{ceil}(\sqrt{b})$ and $\text{floor}(\sqrt{b^2 - 1}) + 1$? Wait, $\sqrt{b^2 - 1}$ is roughly $b - 1/b$, so the integer $\sqrt{n}$ would be up to $b - 1$. So the squares we are considering are k^2 where k is from $\text{ceil}(\sqrt{b})$ up to $b - 1$. Wait, but maybe it's easier to think that n is a two-digit number in base b , so n is between b and $b^2 - 1$, and n must be a perfect square. So k^2 is between b and $b^2 - 1$, so k is between $\text{ceil}(\sqrt{b})$ and $\text{floor}(\sqrt{b^2 - 1})$ which is $b - 1$. Therefore, the number of two-digit squares in base b is $(b - 1) - \text{ceil}(\sqrt{b}) + 1$.

But not all of these squares are necessarily beautiful. To be beautiful, the sum of the two digits in base b must equal $\sqrt{n}$. So for each k from $\text{ceil}(\sqrt{b})$ to $b - 1$, we need to check if k equals the sum of the digits of k^2 when written in base b .

So the problem reduces to finding the smallest b such that the number of k satisfying this condition is more than ten. So we need to find b where there are more than ten k 's such that when you write k^2 in base b , the sum of its digits is k .

Let me formalize this. Let's denote the two digits of k^2 in base b as d_1 and d_2 , so that $k^2 = d_1 \cdot b + d_2$, where $0 < d_1 < b$ and $0 \leq d_2 < b$. Then, the condition is $d_1 + d_2 = k$.

Therefore, substituting d_1 and d_2 into the equation:

$$d_1 + d_2 = k$$

739

But since $k^2 = d_1 * b + d_2$, we can substitute $d_2 = k - d_1$ into this equation:

$$k^2 = d_1 * b + (k - d_1)$$

So:

$$k^2 = d_1 * b + k - d_1$$

Rearranging:

$$k^2 - k = d_1 * b - d_1$$

Factor out d_1 on the right:

$$k^2 - k = d_1(b - 1)$$

Therefore:

$$d_1 = (k^2 - k)/(b - 1)$$

So d_1 must be an integer between 1 and $b - 1$ (since it's the first digit in base b). Therefore, $(k^2 - k)$ must be divisible by $(b - 1)$, and the result d_1 must satisfy $1 \leq d_1 \leq b - 1$.

Moreover, since $d_2 = k - d_1$, and d_2 must satisfy $0 \leq d_2 < b$. So:

$$0 \leq k - d_1 < b$$

But since $d_1 = (k^2 - k)/(b - 1)$, substitute:

$$0 \leq k - (k^2 - k)/(b - 1) < b$$

Multiply through by $(b - 1)$ (which is positive, so inequalities stay the same):

$$0 \leq (k(b - 1)) - (k^2 - k) < b(b - 1)$$

Simplify the left inequality:

$$k(b - 1) - (k^2 - k) \geq 0$$

$$k(b - 1) - k^2 + k \geq 0$$

Factor k :

$$k(b - 1 + 1) - k^2 \geq 0$$

$$\text{Wait, } k(b - 1) + k - k^2 \geq 0$$

That is:

$$k*b - k^2 \geq 0$$

$$k(b - k) \geq 0$$

Since k is positive (as it's the square root of n , which is a two-digit number, so $k \geq \lceil \sqrt{b} \rceil$), and $b - k$ must be non-negative. Wait, but since k can go up to $b - 1$, then $b - k$ is at least 1. So actually, $k(b - k) \geq 0$ is always true because k and $(b - k)$ are both positive. Therefore, the left inequality $0 \leq \dots$ is automatically satisfied.

Now, the right inequality:

$$(k(b - 1) - (k^2 - k)) < b(b - 1)$$

Simplify the left side:

$$k(b - 1) - k^2 + k = kb - k - k^2 + k = kb - k^2$$

$$\text{等等, } (k \cdot (b - 1) - (k^2 - k)) = k \cdot b - k - k^2 + k = k \cdot b - k^2。$$

So the right inequality is:

$$kb - k^2 < b(b - 1)$$

Rearranged:

- $k^2 + b \cdot k < b^2 - b$

Multiply both sides by -1 (inequality flips):

$$k^2 - b \cdot k > b^2 - b$$

But this seems complicated. 或许更好的方法是将其视为关于 k 的二次不等式:

$$k^2 - b \cdot k - (b^2 - b) < 0$$

嗯, 也许可以重新整理一下:

$$k^2 - b \cdot k + (b^2 - b) > 0$$

等等, 我不太确定。让我再检查一下。

Starting from:

$$k^2 - b \cdot k > -(b^2 - b)$$

Which is:

$$k^2 - b \cdot k + (b^2 - b) > 0$$

But this seems like a quadratic in k . Let's consider:

$$k^2 - b \cdot k + (b^2 - b) > 0$$

The discriminant of this quadratic is:

$$D = b^2 - 4(b^2 - b) = b^2 - 4b^2 + 4b = -3b^2 + 4b$$

But since $b \geq 2$, $-3b^2 + 4b$ is negative for $b \geq 2$ because $-3b^2 + 4b < 0$ when $3b^2 > 4b$, which is true for $b > 4/3$. Since $b \geq 2 > 4/3$, discriminant is negative. Therefore, the quadratic $k^2 - bk + (b^2 - b)$ is always positive for all k (since the coefficient of k^2 is positive and discriminant is negative). Therefore, the inequality $k^2 - bk + (b^2 - b) > 0$

$b) > 0$ is always true for all real k , which would imply that the original inequality $kb - k^2 < b(b - 1)$ is always true.

Wait, but that can't be. Wait, perhaps my algebra is wrong.

741

等等，我们来一步步重新追踪一下。

原来的表达式是：

$$k \cdot b - k^2 < b \cdot (b - 1)$$

将其重新整理为：

$$-k^2 + b \cdot k - b^2 + b < 0$$

将所有项移到左侧，可以表示为：

$$-k^2 + b \cdot k - b^2 + b < 0$$

两边同时乘以 -1（注意不等号方向需要改变）：

$$k^2 - b \cdot k + b^2 - b > 0$$

这是一个关于 k 的二次不等式。我们来计算判别式：

$$D = (-b)^2 - 4 \cdot 1 \cdot (b^2 - b) = b^2 - 4b^2 + 4b = -3b^2 + 4b$$

对于 $b \geq 2$ ，有：

$$-3b^2 + 4b \leq -12 + 8 = -4 < 0$$

因此，判别式为负数，这意味着二次函数 $k^2 - b \cdot k + b^2 - b$ 对于所有 k 都是正的（因为二次项系数为正）。因此，无论 k 取何值，该不等式 $k^2 - b \cdot k + b^2 - b > 0$ 恒成立。

由此可知，最初的不等式 $k \cdot b - k^2 < b \cdot (b - 1)$ 对于所有 k 都成立。但这种情况似乎不太合理，因为如果 k 足够大，比如 $k = b$ ，那么 $k \cdot b - k^2 = 0$ ，这显然小于 $b \cdot (b - 1)$ 。等等，但如果 k 的取值范围是从 $[\sqrt{b}]$ 到 $b - 1$ ，那么 $k \leq b - 1$ 。因此， $k \cdot b - k^2$ 的最大值出现在 $k = \frac{b}{2}$ 时（因为这个表达式相当于

$-k^2 + b \cdot k$ ，这是一条开口向下的抛物线）。其最大值为 $\frac{b^2}{4}$ 。而 $b \cdot (b - 1)$ 等于 $b^2 - b$ 。当 $b \geq 2$ 时，显然有 $\frac{b^2}{4} < b^2 - b$ ，这可以通过以下方式验证：

$$\frac{b^2}{4} < b^2 - b \implies 0 < \frac{3b^2}{4} - b \implies 0 < 3b^2 - 4b.$$

\]

对于 $b \geq 2$, 有 $3b^2 - 4b \geq 12 - 8 = 4 > 0$ 。因此, 不等式成立。由此可知, 对于所有 k , 包括 k 在区间 $[\sqrt{b}, b-1]$ 内的所有值, 该不等式 $k \cdot b - k^2 < b \cdot (b-1)$ 始终成立。因此, 右侧的不等式总是成立的。

由此可以得出, $d_2 = k - d_1$ 总是满足 $0 \leq d_2 < b$ 。因此, 我们只需要考虑 $d_1 = \frac{k^2 - k}{b-1}$ 是一个整数且满足 $1 \leq d_1 \leq b-1$ 的 k 即可。

Therefore, the problem reduces to finding the number of integers k such that:

1. $\text{ceil}(\sqrt{b}) \leq k \leq b-1$
2. $(k^2 - k)$ is divisible by $(b-1)$
3. $(k^2 - k)/(b-1)$ is between 1 and $b-1$ inclusive.

So the key is that $(k^2 - k)$ must be divisible by $(b-1)$, and the result d_1 must be a valid digit in base b .

Therefore, for each k in the range $\text{ceil}(\sqrt{b})$ to $b-1$, check if $(k^2 - k)$ is divisible by $(b-1)$, and if the resulting d_1 is between 1 and $b-1$.

But since $d_1 = (k^2 - k)/(b-1)$, and d_1 must be at least 1, so $(k^2 - k)/(b-1) \geq 1 \Rightarrow k^2 - k \geq b-1 \Rightarrow k^2 - k - b + 1 \geq 0$.

Similarly, $d_1 \leq b-1 \Rightarrow (k^2 - k)/(b-1) \leq b-1 \Rightarrow k^2 - k \leq (b-1)^2$.

Therefore, for each k , we have:

$$k^2 - k - b + 1 \geq 0$$

and

$$k^2 - k \leq (b-1)^2$$

But perhaps these conditions are automatically satisfied given the range of k ?

等等, 我们分析一下第一个条件: $k^2 - k \geq b-1$ 。因为 k 至少是 $\text{ceil}(\sqrt{b})$, 我们来看当 $k = \text{ceil}(\sqrt{b})$ 时, $k^2 - k$ 的值是多少。如果 $k \geq \sqrt{b}$, 那么 $k^2 \geq b$, 因此 $k^2 - k \geq b - k$ 。由于 $k \leq b-1$ (因为 k 最大到 $b-1$), 所以 $b - k \geq 1$ 。因此, $k^2 - k \geq b - k \geq 1$ 。但我们需要的是 $k^2 - k \geq b-1$ 。嗯, 当 $k = \text{ceil}(\sqrt{b})$ 时, $k^2 - k \geq b-1$ 吗?

我们来看一个例子, 令 $b = 4$ 。那么 $\text{ceil}(\sqrt{4}) = 2$ 。此时 $k = 2$, 则 $k^2 - k = 4 - 2 = 2$ 。而 $b-1 = 3$ 。所以 $2 \geq 3$ 并不成立。因此, 对于 $b = 4$, 当 $k = 2$ 时, $k^2 - k = 2 < 3 = b-1$ 。因此, $d_1 = (2)/(4-1) = 2/3$, 这不是整数。所以 $k = 2$ 不符合条件。因此, 第一个条件 $k^2 - k \geq b-1$ 并不总是满足, 因此我们需要确保这一条件成立。

Similarly, the second condition $k^2 - k \leq (b-1)^2$. Let's check when $k = b-1$. Then $k^2 - k = (b-1)^2 - (b-1) = (b-1)(b-2)$. Which is equal to $(b-1)^2 - (b-1)$. Wait, but $(b-1)^2$ is $(b^2 - 2b + 1)$. So $k^2 - k = (b-1)^2 - (b-1) + (b-1) = (b-1)^2$. Wait, no:

等一下, $k^2 - k = (b - 1)^2 - (b - 1)$? 等等, $(b - 1)^2 = b^2 - 2b + 1$ 。而 $(b - 1)^2 - (b - 1) = b^2 - 2b + 1 - b + 1 = b^2 - 3b + 2$ 。但这并不是 $k^2 - k$ 。等等, 或许我需要换一种思路。

743

Wait, when $k = b - 1$, $k^2 - k = (b - 1)(b - 1 - 1) = (b - 1)(b - 2) = b^2 - 3b + 2$. On the other hand, $(b - 1)^2 = b^2 - 2b + 1$. So $k^2 - k = b^2 - 3b + 2$. Is this less than or equal to $(b - 1)^2 = b^2 - 2b + 1$? Let's compare:

$$b^2 - 3b + 2 \leq b^2 - 2b + 1$$

Subtract b^2 from both sides:

- $3b + 2 \leq -2b + 1$

Add $2b$ to both sides:

- $b + 2 \leq 1$

Subtract 2:

- $b \leq -1$

Multiply by -1 :

$$b \geq 1$$

Which is always true since $b \geq 2$. Therefore, $k^2 - k \leq (b - 1)^2$ for all $k \leq b - 1$ and $b \geq 2$. Therefore, the second condition is always satisfied. Therefore, the only condition we need to check is that $(k^2 - k)$ is divisible by $(b - 1)$ and that $d1 = (k^2 - k)/(b - 1)$ is at least 1. Wait, but earlier with $b=4$, $k=2$, $d1=2/3$ which is not an integer, so even if $d1$ is supposed to be at least 1, if it's not integer, it's invalid. Therefore, the main conditions are:

1. $(k^2 - k)$ is divisible by $(b - 1)$
2. $(k^2 - k)/(b - 1) \geq 1$

Which translates to $k^2 - k \geq b - 1$.

Therefore, combining these, for each k from $\text{ceil}(\sqrt{b})$ to $b - 1$, we need:

$(k^2 - k)$ is divisible by $(b - 1)$

and

$$k^2 - k \geq b - 1.$$

But since $k \geq \text{ceil}(\sqrt{b})$, let's check if $k^2 - k \geq b - 1$.

If $k \geq \sqrt{b}$, then $k^2 \geq b$, so $k^2 - k \geq b - k$. But since $k \geq 2$ (because $b \geq 2$, $\text{ceil}(\sqrt{b}) \geq 2$), so if $k \geq 2$, then $b - k \leq b - 2$. Wait, but we need $k^2 - k \geq b - 1$. For example, take $b=4$, $\text{ceil}(\sqrt{4})=2$. Then $k=2$, $k^2 - k=2$, which is $2 \geq 3$? No. So in this case, $k^2 - k=2 < b - 1=3$. So this condition is not satisfied. Therefore, even though $k \geq \text{ceil}(\sqrt{b})$, $k^2 - k$ might still be less than $b - 1$. Therefore, we need to ensure that $k^2 - k \geq b - 1$. Let's see when this happens.

Rearranged:

$$k^2 - k - b + 1 \geq 0$$

This is a quadratic in k. Let's solve for k:

$$k^2 - k - (b - 1) \geq 0$$

The roots of the equation $k^2 - k - (b - 1) = 0$ are:

$$k = [1 \pm \sqrt{1 + 4(b - 1)}]/2 = [1 \pm \sqrt{4b - 3}]/2$$

So the positive root is $[1 + \sqrt{4b - 3}]/2$. Therefore, k must be greater than or equal to this value for $k^2 - k - (b - 1) \geq 0$.

Therefore, the minimal k required is $\text{ceil}([1 + \sqrt{4b - 3}]/2)$

Therefore, the range of k is from $\max(\text{ceil}(\sqrt{b}), \text{ceil}([1 + \sqrt{4b - 3}]/2))$ to $b - 1$.

But this seems a bit complicated. Alternatively, perhaps it's easier to note that for each k, we can compute $d_1 = (k^2 - k)/(b - 1)$. If d_1 is an integer and $1 \leq d_1 \leq b - 1$, and $0 \leq d_2 = k - d_1 < b$, then $n = k^2$ is b-beautiful.

But since $d_2 = k - d_1$, and $d_1 = (k^2 - k)/(b - 1)$, then $d_2 = k - (k^2 - k)/(b - 1) = [k(b - 1) - k^2 + k]/(b - 1) = [kb - k^2]/(b - 1)$.

But we already saw that this is equal to $(kb - k^2)/(b - 1)$. Wait, but since $n = k^2$ is two digits in base b, then $d_1b + d_2 = k^2$. Therefore, d_2 must be equal to $k^2 - d_1b$. But d_2 is also equal to $k - d_1$. Therefore, from $n = k^2 = d_1b + d_2$, and $d_2 = k - d_1$, so substituting gives $k^2 = d_1b + (k - d_1) = d_1(b - 1) + k$. Therefore, $d_1 = (k^2 - k)/(b - 1)$. So as before.

Therefore, the key is that $(k^2 - k)$ must be divisible by $(b - 1)$, and $d_1 = (k^2 - k)/(b - 1)$ must be between 1 and $b - 1$ inclusive, and $d_2 = k - d_1$ must be between 0 and $b - 1$ inclusive. But since $d_1 = (k^2 - k)/(b - 1)$, and $d_2 = k - d_1$, then $d_2 = [k(b - 1) - k^2 + k]/(b - 1) = (kb - k^2)/(b - 1)$. But since $n = k^2$ is two digits, d_2 must be less than b. However, since $d_2 = (kb - k^2)/(b - 1)$, we need to ensure that this is non-negative and less than b. But as we saw earlier, the right inequality is always satisfied because the quadratic is always negative. Wait, perhaps I need to check that d_2 is non-negative.

So $d_2 = (kb - k^2)/(b - 1) \geq 0 \Rightarrow kb - k^2 \geq 0 \Rightarrow k(b - k) \geq 0$. Since $k > 0$ and $b - k \geq 0$ (because $k \leq b - 1$), this is always true. Therefore, d_2 is non-negative. And since $d_2 = k - d_1$, and $d_1 \geq 1$, then $d_2 \leq k - 1 < b - 1$ (since $k \leq b - 1$). 等等, 如果 $k = b - 1$, 那么 $d_2 = (b - 1) - d_1$. 而 $d_1 = (k^2 - k)/(b - 1) = [(b - 1)^2 - (b - 1)]/(b - 1) = [b^2 - 2b + 1 - b + 1]/(b - 1) = [b^2 - 3b + 2]/(b - 1) = (b - 1)(b - 2)/(b - 1) = b - 2$. 因此, $d_2 = (b - 1) - (b - 2) = 1$, 这个值小于 b. 所以, d_2 是有效的。

Therefore, the only conditions are:

1. $(k^2 - k)$ is divisible by $(b - 1)$
2. $d_1 = (k^2 - k)/(b - 1)$ is between 1 and $b - 1$ inclusive.

Therefore, for each k from $\text{ceil}(\sqrt{b})$ to $b - 1$, check these two conditions. The number of such k's is the number of b-beautiful numbers.

Therefore, the problem reduces to finding the smallest $b \geq 2$ such that the number of k satisfying these conditions is more than 10.

Therefore, we need to find the minimal b where there are more than ten divisors $(k^2 - k)$ of $(b - 1)$, with d_1 in the valid range.

745

Alternatively, since $(b - 1)$ divides $(k^2 - k)$, we can write $(b - 1)$ divides $k(k - 1)$. Since k and $k - 1$ are consecutive integers, they are coprime, so $(b - 1)$ must divide either k or $(k - 1)$. Therefore, either $k \equiv 0 \pmod{b - 1}$ or $k \equiv 1 \pmod{b - 1}$. Therefore, for each k in the range $\text{ceil}(\sqrt{b})$ to $b - 1$, if $k \equiv 0$ or $1 \pmod{b - 1}$, then $(b - 1)$ divides $k(k - 1)$, hence $(b - 1)$ divides $(k^2 - k)$. Therefore, the possible k 's are those congruent to 0 or 1 modulo $(b - 1)$. However, since k must be in the range $\text{ceil}(\sqrt{b})$ to $b - 1$, and $d_1 = (k^2 - k)/(b - 1)$ must be between 1 and $b - 1$ inclusive.

Therefore, the number of such k 's is equal to the number of integers k in $[\text{ceil}(\sqrt{b}), b - 1]$ such that $k \equiv 0$ or $1 \pmod{b - 1}$ and $d_1 = (k^2 - k)/(b - 1) \leq b - 1$.

Wait, but $(b - 1)$ divides $k(k - 1)$. If $k \equiv 0 \pmod{b - 1}$, then $k = m^*(b - 1)$. Similarly, if $k \equiv 1 \pmod{b - 1}$, then $k = 1 + m^*(b - 1)$. Therefore, since k is in $[\text{ceil}(\sqrt{b}), b - 1]$, the possible values of m are such that $m^*(b - 1) \geq \text{ceil}(\sqrt{b})$ and $1 + m^*(b - 1) \geq \text{ceil}(\sqrt{b})$, and both k 's are less than b .

But since k must be less than b , for $k = m^*(b - 1) < b$, so $m < b/(b - 1)$. Since m is a positive integer, $m = 1$. Similarly, for $k = 1 + m^*(b - 1) < b$, then $m^*(b - 1) < b - 1$, so $m < (b - 1)/(b - 1) = 1$. Hence $m = 0$. Therefore, only possible k 's are:

If $k \equiv 0 \pmod{b - 1}$, then $k = 0$ or $b - 1$. But k cannot be 0 because k is at least $\text{ceil}(\sqrt{b})$. So only possible k is $b - 1$.

But $k \equiv 0 \pmod{b - 1}$ would require $k = 0$ or $k = b - 1$. But $k = 0$ is invalid, so $k = b - 1$. For $k = b - 1$, check if $d_1 = ((b - 1)^2 - (b - 1))/(b - 1) = (b^2 - 2b + 1 - b + 1)/(b - 1) = (b^2 - 3b + 2)/(b - 1) = (b - 1)(b - 2)/(b - 1) = b - 2$. Which is between 1 and $b - 1$.

So $d_1 = b - 2$, $d_2 = k - d_1 = b - 1 - (b - 2) = 1$. So $n = (b - 1)^2$. Let's check if this is a two-digit number in base b . $(b - 1)^2$ in base b is a two-digit number: the first digit is $d_1 = b - 2$, the second digit is 1. So yes, it's a two-digit number. So $k = b - 1$ is always a solution.

Similarly, for $k \equiv 1 \pmod{b - 1}$, then $k = 1 + m^*(b - 1)$. Since $k \geq \text{ceil}(\sqrt{b})$, and $k < b$. So $m = 0$ gives $k = 1$, but 1 is only valid if $\text{ceil}(\sqrt{b}) \leq 1$, which is only when $b = 2$ (since $\sqrt{2} \approx 1.41$, ceil is 2, but $b \geq 2$). 当 $(b = 2)$ 时, $\lceil \sqrt{2} \rceil = 2$, 因此 $(k = 1)$ 不成立。当 $(b \geq 3)$ 时, $\lceil \sqrt{b} \rceil \geq 3$, 所以 $(k = 1)$ 也不成立。因此, 只有 $(m = 0)$ 时, $(k = 1)$ 是可能的, 但仅在 $(\lceil \sqrt{b} \rceil \leq 1)$ 的情况下才成立, 这只有当 $(b = 2)$ 时才有可能。然而, 当 $(b = 2)$ 时, $\lceil \sqrt{2} \rceil = 2$, 所以 $(k = 1)$ 并不符合条件。因此, 对于 $(k \equiv 1 \pmod{b - 1})$, 唯一可能的 $(k = 1)$ 被排除在外, 除非 $(b = 2)$ 。但即便如此, $(k = 1)$ 也不符合条件。因此, $(k \equiv 1 \pmod{b - 1})$ 并没有在范围内的解。等等, 或许我算错了。

等一下, 如果 $(k \equiv 1 \pmod{b - 1})$, 那么 $(k = 1 + m \cdot (b - 1))$, 其中 $(m \geq 0)$ 。但 (k) 必须满足 $(\text{ceil}(\sqrt{b}) \leq k \leq b - 1)$ 之间。如果 $(m = 0)$, 那么 $(k = 1)$ 。只有当 $(\text{ceil}(\sqrt{b}) \leq 1)$ 时, $(k = 1)$ 才有效, 但这只有在 $(b = 2)$ 时才可能成立。然而, 当 $(b = 2)$ 时, $(\text{ceil}(\sqrt{2}) = 2)$, 因此 $(k = 1)$ 并不符合条件。

如果 $(m = 1)$, 那么 $(k = 1 + (b - 1) = b)$ 。然而 (k) 必须小于 (b) , 所以 $(k = b)$ 超出了范围。

因此, 对于 $(k \equiv 1 \pmod{b - 1})$, 没有任何 (k) 位于范围内。所以只有 $(k = b - 1)$ 是符合条件的唯一 (k) 。但这与之前的结论相矛盾。

等一下，我们来看一个例子：假设 $(b = 5)$ 。那么 $(b - 1 = 4)$ 。于是， (k) 必须满足 $(\lceil \sqrt{5} \rceil \approx 3)$ ，所以 $(3 \leq k \leq 4)$ 。接着检查 $(k = 3)$ 和 $(k = 4)$ 。

For $k=3$: $(3^2 - 3)/4 = 6/4 = 1.5$, not integer. So invalid.

For $k=4$: $(16 - 4)/4 = 12/4 = 3$. So $d_1=3$, $d_2=4 - 3=1$. So $n=16=4^2$. In base 5, 16 is 31, which sums to 4. So yes, that's valid. So here, only $k=4$ is valid. So in base 5, there's only one beautiful number. 等等，但根据前面的逻辑， $k=b-1=4$ 是一个解。然而， $k=4 \equiv 0 \pmod{4}$ ？不， $4 \equiv 0 \pmod{4}$ 。所以 $4 \equiv 0 \pmod{4}$ 。因此， $k=4$ 是一个解。但是，在这种情况下， $k=4 \equiv 0 \pmod{4}$ ，所以它是一个解。然而，在 base 5 中， $k=4$ 是唯一的解。因此，对于 base 5，只有一个美丽的数。

等等，这与之前的结论矛盾：之前我认为 $(k \equiv 0 \pmod{(b-1)})$ 或 $(k \equiv 1 \pmod{(b-1)})$ 。然而，当 $(b = 5)$ 时， $(b - 1 = 4)$ ，因此 $(k \equiv 0 \pmod{4})$ 或 $(k \equiv 1 \pmod{4})$ 。在范围 $([3, 4])$ 中， $(k = 4 \equiv 0 \pmod{4})$ ，而 $(k = 3 \equiv 3 \pmod{4})$ 。所以只有 $(k = 4)$ 满足条件。但当 $(k = 4)$ 时， $((k^2 - k) / 4 = (16 - 4)/4 = 12/4 = 3)$ ，这是整数。因此， $(k = 4)$ 是一个有效的解。因此，在基数 $(b = 5)$ 中，只有一个美丽的数。

等等，但也许除了 $(k = b - 1)$ 之外，还有其他可能的 (k) 值。我们来看 $(b = 6)$ 的情况。那么 $(b - 1 = 5)$ 。因此， (k) 必须满足 $(\lceil \sqrt{6} \rceil = 3)$ ，所以 $(3 \leq k \leq 5)$ 。接下来检查 $(k = 3, 4, 5)$ 。

$k=3$: $(9 - 3)/5 = 6/5 = 1.2$, not integer.

$k=4$: $(16 - 4)/5 = 12/5 = 2.4$, not integer.

$k=5$: $(25 - 5)/5 = 20/5 = 4$. So $d_1=4$, $d_2=5 - 4=1$. So $n=25$. In base 6, 25 is 41, which sums to 5. So yes, $\sqrt{25}=5$. So this is valid. So in base 6, only $k=5$ is valid. So again, only one beautiful number.

等等，这表明，只有 $(k = b - 1)$ 是有效的解。那么，为什么题目中会存在超过十个的“美丽”数呢？这好像不对。可能我的推理中存在错误。

等等，我们来看一个不同的例子。假设 $(b = 8)$ 。那么 $(b - 1 = 7)$ 。于是， (k) 的范围是 $(\lceil \sqrt{8} \rceil = 3)$ 到 (7) 。接下来检查 $(k = 3, 4, 5, 6, 7)$ 。

$k=3$: $(9 - 3)/7 = 6/7 \approx 0.857$, not integer.

$k=4$: $(16 - 4)/7 = 12/7 \approx 1.714$, not integer.

$k=5$: $(25 - 5)/7 = 20/7 \approx 2.857$, not integer.

$k=6$: $(36 - 6)/7 = 30/7 \approx 4.285$, not integer.

$k=7$: $(49 - 7)/7 = 42/7 = 6$. So $d_1=6$, $d_2=7 - 6=1$. So $n=49=7^2$. In base 8, 49 is 61, which sums to 7. Correct. So only $k=7$ is valid. So again, only one.

等等，这表明对于每个 b ，只有 $k = b - 1$ 是有效的？如果是这样，那么每个 b 只会有一个 b -beautiful 数，但这与题目要求矛盾，题目中提到需要有超过十个 b -beautiful 数。因此，我的推理一定存在错误。

等等，或许还有其他的 k 满足条件 $(k^2 - k)$ 可以被 $(b - 1)$ 整除。之前我假设有 $k \equiv 0 \pmod{(b-1)}$ 或 $k \equiv 1 \pmod{(b-1)}$ ，但这种情况仅适用于 $(b - 1)$ 是质数的时候。如果 $(b - 1)$ 不是质数，那么可能会有更多的 k 满足条件 $(k^2 - k)$ 能被 $(b - 1)$ 整除。

Wait, for example, suppose $b-1$ is composite. Let's take $b=7$, so $b-1=6$. Then, since 6 divides $k(k-1)$. Since k and $k-1$ are coprime, 6 must divide either k or $k-1$. Therefore, either $k \equiv 0 \pmod{6}$ or $k \equiv 1 \pmod{6}$. So in base 7, the possible k 's are those congruent to 0 or 1 mod 6. The range for k is $\text{ceil}(\sqrt{7})=3$ to 6. So $k=3,4,5,6$.

Check $k=3$: $3 \bmod 6 = 3 \neq 0$ or 1. So invalid.

$k=4$: $4 \bmod 6 = 4 \neq 0$ or 1. Invalid.

$k=5$: $5 \bmod 6 = 5 \neq 0$ or 1. Invalid.

$k=6$: $6 \equiv 0 \pmod{6}$. So valid. So $k=6$. Then $d_1=(36-6)/6=30/6=5$. So $d_1=5$, $d_2=6-5=1$. So $n=36=6^2$. In base 7, 36 is $5 \cdot 7 + 1 = 35 + 1 = 36$. So $51_{\text{base } 7}$, which sums to 6. Correct. So only $k=6$ is valid here. So again, only one.

等等，即使 $(b-1)$ 是合数，满足条件的 (k) 似乎仍然是唯一的。这表明，对于每个基底 (b) ，只有 $(k = b-1)$ 是满足条件的。但这与题目描述矛盾，因为题目中提到 (81) 是 (13) -beautiful，且存在其他 (b) -beautiful 的数字。等等，也许我错在假设只有 $(k \equiv 0 \pmod{b-1})$ 或 $(k \equiv 1 \pmod{b-1})$ 才能满足条件。让我再仔细想一想。

If $(b-1)$ divides $k(k-1)$, since k and $k-1$ are coprime, $(b-1)$ must divide either k or $k-1$. Therefore, either $(b-1)$ divides k or $(b-1)$ divides $k-1$. Therefore, $k \equiv 0 \pmod{b-1}$ or $k \equiv 1 \pmod{b-1}$. However, since k is in $[\text{ceil}(\sqrt{b}), b-1]$, the number of such k 's is equal to the number of multiples of $(b-1)$ in that interval (which is 1: $k=b-1$) plus the number of numbers congruent to 1 mod $(b-1)$ in that interval. Similarly, if $(b-1)$ divides $k-1$, then $k \equiv 1 \pmod{b-1}$. So numbers congruent to 1 mod $(b-1)$ in the interval.

But in the interval $[\text{ceil}(\sqrt{b}), b-1]$, how many numbers are congruent to 0 or 1 mod $(b-1)$?

For example, take $b=10$. Then $b-1=9$. The interval is $\text{ceil}(\sqrt{10})=4$ to 9. So numbers 4,5,6,7,8,9.

Numbers congruent to 0 mod 9: 9. Numbers congruent to 1 mod 9: 1,10. But 1 is less than $\text{ceil}(\sqrt{10})=4$, and 10 is equal to $b=10$, which is excluded. So only $k=9 \equiv 0 \pmod{9}$. So here, only $k=9$. Then $d_1=(81-9)/9=72/9=8$. So $d_1=8$, $d_2=9-8=1$. So $n=81=9^2$. In base 10, 81 is 81, which sums to 9. Correct. So only one beautiful number here.

等等，这表明在每个基数中都只有一个美丽的数字。但题目中提到，需要找到一个 b ，使得美丽的数字超过十个。这显然与我的推导相矛盾。因此，我的推理中一定存在某种错误。

Wait, let's take the example given: 81 is 13-beautiful. So in base 13, 81 is written as 63. Let's check if $6+3=9=\sqrt{81}$. Correct. So $k=9$. So in base 13, $k=9$. Let's check if (9^2-9) is divisible by 12. $81-9=72$. 72 divided by 12 is 6, which is integer. So $d_1=72/12=6$, $d_2=9-6=3$. So yes, in base 13, 81 is $6 \cdot 13 + 3 = 78 + 3 = 81$. So $63_{\text{base } 13}$. So digits 6 and 3 sum to 9. Therefore, this works. So here, $k=9$. For base 13, $b-1=12$. So 9 is neither 0 nor 1 mod 12. So $9 \not\equiv 0 \pmod{12}$. But (12) divides $(9^2-9)=72$. Indeed, 12 divides 72. So even though 9 is not congruent to 0 or 1 mod 12, 12 divides 72. Therefore, my earlier assumption that k must be $\equiv 0$ or 1 mod $(b-1)$ is wrong. Because $(b-1)$ divides $k(k-1)$ does not require $k \equiv 0$ or 1 mod $(b-1)$. It only requires that $(b-1)$ divides the product $k(k-1)$. So $(b-1)$ can divide $k(k-1)$ even if it doesn't divide either k or $k-1$, as long as the product is divisible. For example, if $(b-1)$ is composite, its factors could divide k and $k-1$ respectively.

Therefore, my previous reasoning that k must be $\equiv 0$ or 1 mod $(b-1)$ is incorrect. Therefore, the number of k 's such that $(b-1)$ divides $k(k-1)$ is not limited to those congruent to 0 or 1 mod $(b-1)$. Therefore, the number of such k 's can

be higher. Therefore, perhaps for some bases b , there are multiple k 's where $(b-1)$ divides $k(k-1)$. Therefore, the number of b -beautiful numbers can be more than one. Therefore, the problem is to find the minimal b where there are more than ten such k 's.

Therefore, the approach should be:

For a given b , find all integers k in $[\text{ceil}(\sqrt{b}), b-1]$ such that $(b-1)$ divides $(k^2 - k)$. Then count these k 's. We need to find the minimal b where this count exceeds ten.

But how to compute this efficiently? Since b can be large, maybe we need a formula or a pattern.

Alternatively, note that $(b-1)$ divides $(k^2 - k)$ can be written as $k^2 \equiv k \pmod{b-1}$. So $k^2 \equiv k \pmod{b-1}$. Therefore, $k(k-1) \equiv 0 \pmod{b-1}$. So the number of k 's in $[\text{ceil}(\sqrt{b}), b-1]$ satisfying this congruence.

This is equivalent to solving $k(k-1) \equiv 0 \pmod{b-1}$ for k in the given range. The number of solutions to this congruence is equal to the number of integers k in $[\text{ceil}(\sqrt{b}), b-1]$ such that $k \equiv 0$ or $1 \pmod{d}$ for some divisors d of $(b-1)$. But this might be complicated.

Alternatively, note that for each k in $[\text{ceil}(\sqrt{b}), b-1]$, $k(k-1)$ must be divisible by $(b-1)$. This is equivalent to $(b-1)$ divides $k(k-1)$. So for each k , check if $k(k-1)$ is divisible by $(b-1)$. The number of such k 's is the number of b -beautiful numbers.

Therefore, the problem reduces to, for each b , compute the number of k in $[\text{ceil}(\sqrt{b}), b-1]$ such that $(b-1)$ divides $k(k-1)$. We need the minimal b where this count exceeds ten.

To find such b , we can start from $b=2$ and increment b until the count exceeds ten. However, this might be time-consuming. Is there a pattern or a way to estimate how many k 's satisfy this for a given b ?

Note that $(b-1)$ divides $k(k-1)$ implies that for each prime power p^r dividing $(b-1)$, p^r divides $k(k-1)$. Since k and $k-1$ are coprime, for each prime p dividing $(b-1)$, either p^r divides k or p^r divides $(k-1)$. Therefore, the number of solutions $k \pmod{b-1}$ is 2^ω , where ω is the number of distinct prime factors of $(b-1)$. However, this is only if $(b-1)$ is square-free. If $(b-1)$ has square factors, the number of solutions can be less.

But in general, the number of solutions to $k(k-1) \equiv 0 \pmod{m}$ is multiplicative over the prime powers in the factorization of m . So if $m = \prod p_i^{r_i}$, then the number of solutions is $\prod (d(p_i^{r_i}))$, where $d(p^r) = 2$ if $p=2$, and 2 otherwise. Wait, no. For each prime power p^r dividing m , the number of solutions to $k(k-1) \equiv 0 \pmod{p^r}$ is:

- If $p=2$, there are 1 or 3 solutions mod 8, but in general for higher powers, it's 2 solutions.

等等，对于奇素数 (p) ，方程 $(k(k-1) \equiv 0 \pmod{p^r})$ 的解是 $(k \equiv 0 \pmod{p^r})$ 或 $(k \equiv 1 \pmod{p^r})$ 。因此，对于每个奇素数幂 (p^r) ，方程有 (2) 个解。

For $p=2$, if $r \geq 3$, then there are 4 solutions mod 8, but for $r=1$, 2 solutions, for $r=2$, 4 solutions. Wait, let's check:

For $p=2$:

If $r=1$: mod 2. The solutions are $k \equiv 0$ or $1 \pmod{2}$. So 2 solutions.

If $r=2$: mod 4. The solutions are $k \equiv 0$ or $1 \pmod{4}$. So 2 solutions.

If $r=3$: $\text{mod } 8$. The equations $k(k-1) \equiv 0 \pmod{8}$. The solutions are $k \equiv 0, 1 \pmod{8}$. Because if k is even, $k-1$ is odd, so 8 divides k . If k is odd, $k-1$ is even, so 8 divides $k-1$. Wait, but 8 divides k or $k-1$. So solutions are $k \equiv 0$ or $1 \pmod{8}$. So 2 solutions. Wait, but 8 divides $k(k-1)$. If k is even, then k must be divisible by 8, because $k-1$ is odd and can't contribute any factors of 2. Similarly, if k is odd, $k-1$ is even and must be divisible by 8. Therefore, solutions are $k \equiv 0$ or $1 \pmod{8}$. So 2 solutions. Similarly, for higher r , the number of solutions is 2. So for $p=2$, regardless of r , the number of solutions is 2.

Therefore, for a general modulus m , the number of solutions to $k(k-1) \equiv 0 \pmod{m}$ is 2^ω , where ω is the number of distinct prime factors of m . However, this is only if m is square-free. If m has square factors, the number of solutions can be less. Wait, but in our case, $m = b-1$. So if $(b-1)$ is square-free, the number of solutions $\text{mod } (b-1)$ is 2^ω . Therefore, the number of solutions k in $[0, b-1]$ is 2^ω . Therefore, the total number of solutions in $[1, b-1]$ is 2^ω . However, we are interested in k in $[\text{ceil}(\sqrt{b}), b-1]$. Therefore, the number of solutions is the number of residues $\text{mod } (b-1)$ that are in $[\text{ceil}(\sqrt{b}), b-1]$ and satisfy $k \equiv 0$ or $1 \pmod{p^r}$ for each prime power in $(b-1)$.

But this is getting complicated. However, if $(b-1)$ is a product of distinct primes (square-free), then the number of solutions $\text{mod } (b-1)$ is 2^ω , where ω is the number of distinct prime factors. Therefore, the number of solutions in the entire range $[0, b-1]$ is 2^ω . Therefore, if we can find b such that $(b-1)$ has many distinct prime factors, then 2^ω will be large, leading to more solutions. Therefore, perhaps choosing b such that $(b-1)$ is a product of the first few primes will lead to a large number of solutions.

For example, let's take $(b-1) = 235 \cdot 7 = 210$. Then $b = 211$. Then the number of solutions $\text{mod } 210$ is $2^4 = 16$. Therefore, in the interval $[0, 210)$, there are 16 solutions. Therefore, in $[\text{ceil}(\sqrt{211}), 210]$, the number of solutions would be 16 minus those solutions less than $\text{ceil}(\sqrt{211}) \approx 15$. So $16 - \text{number of solutions less than } 15$. If none of the solutions are less than 15, then we have 16 solutions. But since the solutions are congruent to 0 or 1 $\text{mod } 2, 3, 5, 7$, which would spread out the residues. However, some of these solutions might be less than 15, but perhaps most are above. Therefore, such a $b = 211$ might have many beautiful numbers. But we need to find the minimal b .

But since the problem asks for the least b where there are more than ten beautiful numbers, we need to find the smallest b such that the number of solutions to $k(k-1) \equiv 0 \pmod{(b-1)}$ in $[\text{ceil}(\sqrt{b}), b-1]$ exceeds ten.

Therefore, to find such b , we can start from small b and check the count. However, given that the example given is 81 being 13-beautiful, and in base 13, $k=9$. Let's check for $b=13$. Then $(b-1) = 12 = 2^2 \cdot 3$. So the number of solutions $\text{mod } 12$ would be $2^2 = 4$? Wait, but 12 is not square-free. For modulus with repeated prime factors, the number of solutions can be different.

Wait, for modulus $m = 12 = 2^2 \cdot 3$. Let's compute the number of solutions to $k(k-1) \equiv 0 \pmod{12}$.

This requires that 4 divides $k(k-1)$ and 3 divides $k(k-1)$. Since k and $k-1$ are coprime:

For 4 divides $k(k-1)$: Since k and $k-1$ are consecutive, one of them is even. But to have 4 divides $k(k-1)$, either k is divisible by 4 or $k-1$ is divisible by 4. Because if k is even, then $k-1$ is odd, so 4 divides k . If k is odd, then $k-1$ is even, and we need 4 divides $(k-1)$. Therefore, solutions $\text{mod } 4$ are $k \equiv 0$ or $1 \pmod{4}$.

Similarly, for 3 divides $k(k-1)$: Either $k \equiv 0 \pmod{3}$ or $k \equiv 1 \pmod{3}$.

Therefore, combining these using the Chinese Remainder Theorem, the solutions mod12 are the combinations of the solutions mod4 and mod3.

For mod4: solutions $k \equiv 0, 1 \pmod{4}$.

For mod3: solutions $k \equiv 0, 1 \pmod{3}$.

Therefore, combining these, we have four combinations:

1. $k \equiv 0 \pmod{4}$ and $k \equiv 0 \pmod{3}$: solution is $k \equiv 0 \pmod{12}$.
2. $k \equiv 0 \pmod{4}$ and $k \equiv 1 \pmod{3}$: solution is $k \equiv 8 \pmod{12}$ (since $8 \equiv 0 \pmod{4}$ and $8 \equiv 2 \pmod{3}$? Wait, $8 \pmod{3} = 2$, so $8 \equiv -1 \pmod{3}$. Wait, we need $k \equiv 0 \pmod{4}$ and $k \equiv 1 \pmod{3}$. Let's solve:

$$k \equiv 0 \pmod{4}$$

$$k \equiv 1 \pmod{3}$$

Let $k=4m$. Then $4m \equiv 1 \pmod{3} \Rightarrow m \equiv 1 \pmod{3}$. So $m=3n+1$. Therefore, $k=4*(3n+1)=12n+4$. Therefore, $k \equiv 4 \pmod{12}$.

等等, $4 \pmod{3} = 1$, 所以 $4 \equiv 1 \pmod{3}$. 由此可得, $k=4$ 满足条件。因此, 解为 $k \equiv 4 \pmod{12}$ 。

Similarly,

3. $k \equiv 1 \pmod{4}$ and $k \equiv 0 \pmod{3}$: $k \equiv 1 \pmod{4}$, $k \equiv 0 \pmod{3}$.

$k=4m+1$. Then $4m+1 \equiv 0 \pmod{3} \Rightarrow 4m \equiv -1 \pmod{3} \Rightarrow m \equiv 2 \pmod{3}$ (since $4 \equiv 1 \pmod{3}$, so $1m \equiv -1 \pmod{3} \Rightarrow m \equiv 2 \pmod{3}$). Therefore, $m=3n+2$. Therefore, $k=4(3n+2)+1=12n+9$. Therefore, $k \equiv 9 \pmod{12}$.

4. $k \equiv 1 \pmod{4}$ and $k \equiv 1 \pmod{3}$: $k \equiv 1 \pmod{4}$, $k \equiv 1 \pmod{3}$. Then $k \equiv 1 \pmod{12}$ (since if $k \equiv 1 \pmod{4}$ and $\pmod{3}$, then $k \equiv 1 \pmod{\text{lcm}(4,3)=12}$).

Therefore, the four solutions mod12 are $k \equiv 0, 1, 4, 9 \pmod{12}$.

Therefore, in the interval $[\text{ceil}(\sqrt{13}), 12]$, which is $[4, 12]$, the valid k's are:

$$k=4, 5, 6, 7, 8, 9, 10, 11, 12.$$

Check which of these are $\equiv 0, 1, 4, 9 \pmod{12}$.

$$k=4: 4 \equiv 4 \pmod{12}. \text{ Valid.}$$

$$k=5: 5 \equiv 5 \pmod{12}. \text{ Not valid.}$$

等等, $k=4 \equiv 4 \pmod{12}$, 符合条件。

$$k=5 \equiv 5 \pmod{12}, \text{ 不符合条件。}$$

$$k=6 \equiv 6 \pmod{12}, \text{ 不符合条件。}$$

$$k=7 \equiv 7 \pmod{12}, \text{ 不符合条件。}$$

$$k=8 \equiv 8 \pmod{12}, \text{ 符合条件。}$$

$$k=9 \equiv 9 \pmod{12}, \text{ 符合条件。}$$

$k=10 \equiv 10 \pmod{12}$, 不符合条件。

$k=11 \equiv 11 \pmod{12}$, 不符合条件。

751

$k=12 \equiv 0 \pmod{12}$, 符合条件。

Therefore, the valid k 's are 4, 8, 9, 12. Now check if these k 's lead to valid $d_1 = (k^2 - k)/12$.

$k=4$:

$(16 - 4)/12 = 12/12 = 1$. $d_1 = 1$, $d_2 = 4 - 1 = 3$. So $n = 4^2 = 16$. In base 13, 16 is $1 \cdot 13 + 3 = 16$. Digits 1 and 3 sum to 4. Correct.

$k=8$:

$(64 - 8)/12 = 56/12 \approx 4.666$. Not integer. So invalid.

等等, 56 并不能被 12 整除。等等, $64 - 8 = 56$ 。 $56 \div 12 \approx 4.666\dots$ 。这不是整数。所以 $k=8$ 是无效的。但根据之前的计算, $k=8$ 满足模 12 的条件。然而实际上, $(k^2 - k)$ 必须能被 12 整除。因此, 尽管 $k=8$ 满足同余条件, 但 56 并不能被 12 整除, 因此 $k=8$ 是无效的。这是一个问题。

等等, 我们来检查一下当 $(k \equiv 8 \pmod{12})$ 的情况: $(k \equiv 8 \pmod{12})$ 。那么 $(k(k-1) = 8 \cdot 7 = 56)$ 。 $(56 \pmod{12} = 56 - 4 \cdot 12 = 56 - 48 = 8)$ 。因此, $(56 \equiv 8 \pmod{12})$ 。所以, (56) 并不被 (12) 整除。然而, 根据之前的分析, $(k \equiv 8 \pmod{12})$ 应该是一个解。但这似乎与事实矛盾。

等等, 也许我的方法存在错误。等一下, 根据之前的分析, 我们考虑的是模 4 和模 3 的条件。对于 $(k=8)$:

$[$

$k \equiv 0 \pmod{4}$ { (因为 $(8 \div 4 = 2)$, 余数为 (0)) },

$k - 1 = 7 \equiv 1 \pmod{3}$ { (因为 $(7 \div 3 = 2)$, 余数为 (1)) }。

$]$

因此, $(k=8)$ 满足 $(k \equiv 0 \pmod{4})$ 且 $(k \equiv 1 \pmod{3})$ 的条件, 这与我们之前计算的第 2 种情况相符。但是实际上, $(8 \cdot 7 = 56)$, 而 (56) 显然不能被 (12) 整除。这是为什么?

等等, (4) 可以整除 (8) , 而 (3) 可以整除 (7) 吗? (7) 不能被 (3) 整除, 但 $(7 \equiv 1 \pmod{3})$ 。等等, (3) 可以整除 $(k(k-1))$ 当且仅当 $(k \equiv 0 \pmod{3})$ 或 $(k \equiv 1 \pmod{3})$ 。由于 $(k=8 \equiv 2 \pmod{3})$, $(k-1=7 \equiv 1 \pmod{3})$ 。因此, (3) 可以整除 $(k-1)$ 。所以 (3) 确实可以整除 (7) , 但这不可能, 除非 (7) 能被 (3) 整除。显然, (7) 不能被 (3) 整除。等等, 我的推理哪里出错了?

等等, $(k(k-1) \equiv 0 \pmod{3})$ 当且仅当 $(k \equiv 0 \pmod{3})$ 或 $(k \equiv 1 \pmod{3})$ 。这是因为 3 是一个素数。所以, 如果 $(k \equiv 2 \pmod{3})$, 那么 $(k-1 \equiv 1 \pmod{3})$, 因此 $(k-1 \equiv 1 \pmod{3})$, 这意味着 (3) 可以整除 $(k-1)$ 。等等, 不对。如果 $(k \equiv 2 \pmod{3})$, 那么 $(k-1 \equiv 1 \pmod{3})$, 所以 (3) 并不能整除 $(k-1)$ 。等等, 如果 $(k \equiv 1 \pmod{3})$, 那么 $(k-1 \equiv 0 \pmod{3})$, 因此 (3) 可以整除 $(k-1)$ 。同样地, 如果 $(k \equiv 0 \pmod{3})$, 那么 (3) 可以整除 (k) 。

因此, 如果 $(k \equiv 2 \pmod{3})$, 则 $(k(k-1) \equiv 2 \cdot 1 = 2 \pmod{3})$, 这不为 0。所以, $(k(k-1) \equiv 0 \pmod{3})$ 当且仅当 $(k \equiv 0 \pmod{3})$ 或 $(k \equiv 1 \pmod{3})$ 。于是, $(k=8 \equiv 2 \pmod{3})$ 。因此, $(8 \cdot 7 \equiv 2 \cdot 1 = 2 \pmod{3})$, 这说明 3 并不能整除 $(8 \cdot 7)$ 。但根据之前的条件, 我们有 $(k \equiv 0 \pmod{4})$ 且 $(k$

$\equiv 1 \pmod 3$ 。然而, $(k=8)$ 满足 $(0 \pmod 4)$, 但 $(8 \equiv 2 \pmod 3)$ 。因此, $(k=8)$ 并不满足 $(1 \pmod 3)$ 的条件。等等, 这与我们的 CRT 解 $(k \equiv 8 \pmod{12})$ 矛盾了。

等等, 我们来看一下。在情况 2 中, 我们要求 $(k \equiv 0 \pmod 4)$ 且 $(k \equiv 1 \pmod 3)$ 。我们找到的解是 $(k \equiv 4 \pmod{12})$ 。等等, 让我们重新解这个问题。

我们设定 $(k \equiv 0 \pmod 4)$ 且 $(k \equiv 1 \pmod 3)$ 。那么 $(k = 4m)$, 其中 (m) 为某个整数。接着, $(4m \equiv 1 \pmod 3)$ 。因为 $(4 \equiv 1 \pmod 3)$, 所以该式可以简化为 $(1 \cdot m \equiv 1 \pmod 3)$ 。因此, $(m \equiv 1 \pmod 3)$ 。由此得到 $(m = 3n + 1)$ 。代入后得到:

$$[\\ k = 4 \cdot (3n + 1) = 12n + 4. \\]$$

因此, $(k \equiv 4 \pmod{12})$ 。

所以, $(k = 4, 16, 28, \dots)$ 。因此, $(k=4)$ 是满足条件的, 但 $(k=8)$ 并不是。等等, $(k=8 \equiv 8 \pmod{12})$ 。所以 (8) 并不符合 $(4 \pmod{12})$ 的条件。

等等, 我现在有些糊涂了。在之前的分析中, 对于模 12 的解, 我们得到了 $(k \equiv 0,1,4,9 \pmod{12})$ 。那么 $(k=4, 8, 9, 12)$ 在区间 $([4,12])$ 中。然而, $(k=8 \equiv 8 \pmod{12})$, 并不在这些解的范围内。等等, 实际上, $(k=8 \equiv 8 \pmod{12})$, 并不满足 $(0,1,4,9 \pmod{12})$ 的条件。因此, 之前的结论是正确的, 解为 $(k=4,9,12)$ 。

等等, $(k=12 \equiv 0 \pmod{12})$ 。但 $(k=12)$ 等于 $(b-1=12)$ 。接下来检查 $(k=12)$:

$$[\\ (144 - 12)/12 = 132/12 = 11. \\]$$

因此, $(d_1=11)$, $(d_2=12 - 11=1)$ 。所以 $(n=144=12^2)$ 。在基底 13 中, (144) 表示为 $(11 \cdot 13 + 1=143 + 1=144)$ 。数字 (11) 和 (1) 的和为 (12) , 而 $(\sqrt{144}=12)$ 。因此, 这确实是成立的。所以 $(k=12)$ 是有效的。

综上, 对于 $(b=13)$, 有效的 (k) 值为 $(4, 9, 12)$ 。现在我们再检查 $(k=9)$:

$$[\\ (81 - 9)/12 = 72/12 = 6. \\]$$

因此, $(d_1=6)$, $(d_2=9 - 6=3)$ 。所以 $(n=81=9^2)$ 。在基底 13 中, $(81=6 \cdot 13 + 3=78 + 3=81)$ 。数字 (6) 和 (3) 的和为 (9) , 而 $(\sqrt{81}=9)$ 。因此, 这也是正确的。

Similarly, $k=4$ as before.

Therefore, in base13, there are three beautiful numbers: 16,81,144, and also $k=12$. 等等, $k=12$ 对应的 $n=144$ 。那么, $k=12$ 表示 $n=144$ 。在基底13中, 144写成111。等等, 144除以13等于11余1, 因此是 $11 \times 13 + 1=143 + 1=144$ 。所以是11和1, 即111。但 $1+1=2$, 这并不等于12。等等, 不对。再等等, $d_1=11$, $d_2=1$, 所以 $11 + 1=12$ 。这确实等于 $\sqrt{144}=12$ 。明白了。所以是正确的。

Therefore, in base13, the beautiful numbers are $n=16, 81, 144$. 等等, 还有 $k=9, 4$, 和 12 。因此, 一共有四个美丽的数字。等等, $k=4$ 对应的 $n=16$, $k=9$ 对应的 $n=81$, $k=12$ 对应的 $n=144$ 。所以总共有三个。等等, 但在区间 $[4, 12]$ 中, $k=4, 9, 12$ 。因此, 一共有三个数。所以 base13 中有三个美丽的数字。然而, 题目中给出的例子是 81 是 13-美丽的, 但这里我们找到了三个。等等, 题目中说: “例如, 81 是 13-美丽的.....”。因此, 可能 base13 有三个这样的数字。

But the problem asks for the least b where there are more than ten such numbers. Therefore, we need to find a larger b . Therefore, perhaps we need to find a b where $(b-1)$ has more prime factors, leading to more solutions.

Alternatively, note that if $(b-1)$ is a product of the first n primes, then the number of solutions is 2^n . For example, if $(b-1)=235711=2310$, then $b=2311$, and the number of solutions would be $2^5=32$. Therefore, in the interval $[\text{ceil}(\sqrt{2311}), 2310]$, the number of solutions would be 32 minus those less than $\text{ceil}(\sqrt{2311}) \approx 48$. So 32 - number of solutions less than 48. If there are, say, 4 solutions less than 48, then we have 28 solutions. But this is speculative. However, this suggests that choosing b such that $(b-1)$ is a product of many primes will lead to a higher number of solutions.

But to find the minimal b , we need to check values of b starting from $b=2$ upwards until we find more than ten solutions. However, this is time-consuming. Let's try to compute for some b .

Let's try $b=30$. Then $(b-1)=29$, which is prime. Therefore, the number of solutions mod 29 is 2. Therefore, in the interval $[\text{ceil}(\sqrt{30}), 29]$, the solutions are $k \equiv 0$ or $1 \pmod{29}$. But in this interval, $k=29 \equiv 0 \pmod{29}$, and $k=1 \equiv 1 \pmod{29}$. But $k=1$ is less than $\text{ceil}(\sqrt{30})=6$, so only $k=29$. Therefore, only one solution. So $n=29^2=841$. In base 30, 841 is $29 \cdot 30 + 1 = 870 + 1 = 871$, which is not correct. Wait, no. Wait, $29^2=841$. In base 30, 841 divided by 30 is $28 \cdot 30 = 840$, so $841 = 28 \cdot 30 + 1$. Therefore, the digits are 28 and 1, which sum to 29. $\sqrt{841}=29$. Correct. So only one beautiful number here.

Similarly, let's try $b=10$. Then $(b-1)=9=3^2$. The number of solutions to $k(k-1) \equiv 0 \pmod{9}$. Since $9=3^2$, we need 9 divides $k(k-1)$. Since k and $k-1$ are coprime, either 9 divides k or 9 divides $k-1$. Therefore, $k \equiv 0$ or $1 \pmod{9}$. In the interval $[\text{ceil}(\sqrt{10}), 9]$, the numbers are 4, 5, 6, 7, 8, 9. Among these, $k=9 \equiv 0 \pmod{9}$. So only $k=9$. Check $k=9$: $(81-9)/9=72/9=8$. $d_1=8$, $d_2=9-8=1$. So $n=81=9^2$. In base 10, $81=8 \cdot 10 + 1$. Digits 8 and 1 sum to 9. Correct. So only one.

Next, try $b=15$. Then $(b-1)=14=2 \cdot 7$. So solutions are $k \equiv 0, 1 \pmod{2}$ and $k \equiv 0, 1 \pmod{7}$. Combining, using Chinese Remainder Theorem:

Solutions mod 14:

1. $k \equiv 0 \pmod{2}$ and $k \equiv 0 \pmod{7}$: $k \equiv 0 \pmod{14}$
2. $k \equiv 0 \pmod{2}$ and $k \equiv 1 \pmod{7}$: $k \equiv 8 \pmod{14}$ (since $8 \equiv 0 \pmod{2}$, $8 \equiv 1 \pmod{7}$)
3. $k \equiv 1 \pmod{2}$ and $k \equiv 0 \pmod{7}$: $k \equiv 7 \pmod{14}$

等等, $7 \pmod{2} = 1$, 对吗? 因为 7 是奇数。所以 $7 \equiv 1 \pmod{2}$ 。因此, $k \equiv 7 \pmod{14}$ 。

4. $k \equiv 1 \pmod{2}$ and $k \equiv 1 \pmod{7}$: $k \equiv 1 \pmod{14}$.

Therefore, solutions are $k \equiv 0, 1, 7, 8 \pmod{14}$. In the interval $[\text{ceil}(\sqrt{15}), 14]$, the numbers are 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14. Check which are $\equiv 0, 1, 7, 8 \pmod{14}$:

$$k=7 \equiv 7 \pmod{14}$$

$$k=8 \equiv 8 \pmod{14}$$

$$k=14 \equiv 0 \pmod{14}$$

$$k=1 \equiv 1 \pmod{14} \text{ (but } 1 < 4, \text{ so excluded)}$$

$k=7, 8, 14$ are valid.

Check $k=7$:

$(49 - 7)/14 = 42/14 = 3$. $d_1=3$, $d_2=7 - 3=4$. $n=49$. In base 15, $49=3 \cdot 15 + 4=45 + 4=49$. Digits 3 and 4 sum to 7. $\sqrt{49}=7$.

Correct.

$k=8$:

$(64 - 8)/14 = 56/14 = 4$. $d_1=4$, $d_2=8 - 4=4$. $n=64$. In base 15, $64=4 \cdot 15 + 4=60 + 4=64$. Digits 4 and 4 sum to 8. $\sqrt{64}=8$.

Correct.

$k=14$:

$(196 - 14)/14 = 182/14 = 13$. $d_1=13$, $d_2=14 - 13=1$. $n=196$. In base 15, $196=13 \cdot 15 + 1=195 + 1=196$. Digits 13 and 1 sum to 14. $\sqrt{196}=14$. Correct.

Thus, in base 15, there are three beautiful numbers: 49, 64, 196. So three solutions. Still not enough.

Next, try $b=22$. Then $(b - 1)=21=3 \cdot 7$. Solutions mod 21:

Solutions are combinations of solutions mod 3 and mod 7.

mod 3: $k \equiv 0$ or $1 \pmod{3}$

mod 7: $k \equiv 0$ or $1 \pmod{7}$

Thus, four combinations:

1. $k \equiv 0 \pmod{3}$ and $k \equiv 0 \pmod{7}$: $k \equiv 0 \pmod{21}$

等等, $0 \pmod{3}$ 是 0, $0 \pmod{7}$ 也是 0。因此, $k \equiv 0 \pmod{21}$ 。

2. $k \equiv 0 \pmod{3}$ and $k \equiv 1 \pmod{7}$: Solve $k=3m=7n+1$. $k \equiv 1 \pmod{7}$ and $k \equiv 0 \pmod{3}$. Let $k=7n+1$. Then $7n+1 \equiv 0 \pmod{3} \Rightarrow 7n \equiv -1 \pmod{3} \Rightarrow n \equiv -1 \pmod{3}$ (since $7 \equiv 1 \pmod{3}$). So $n \equiv 2 \pmod{3}$. Thus, $n=3p+2$. Then $k=7 \cdot (3p+2)+1=21p+15$. Therefore, $k \equiv 15 \pmod{21}$.

3. $k \equiv 1 \pmod{3}$ and $k \equiv 0 \pmod{7}$: $k=3m+1=7n$. So $3m+1=7n$. Then $3m \equiv -1 \pmod{7} \Rightarrow 3m \equiv 6 \pmod{7} \Rightarrow m \equiv 2 \pmod{7}$. So $m=7p+2$. Then $k=3 \cdot (7p+2)+1=21p+7$. Thus, $k \equiv 7 \pmod{21}$.

等等, $3 \cdot (7p+2) + 1 = 21p + 6 + 1 = 21p + 7$ 。因此, $k \equiv 7 \pmod{21}$ 。

4. $k \equiv 1 \pmod{3}$ and $k \equiv 1 \pmod{7}$: $k \equiv 1 \pmod{21}$.

Therefore, solutions are $k \equiv 0, 1, 7, 15 \pmod{21}$.

In the interval $[\text{ceil}(\sqrt{22}) \approx 5, 21]$, the numbers are 5, ..., 21. Check $k \equiv 0, 1, 7, 15, 21 \pmod{21}$.

$k=7, 15, 21 \equiv 0, 7, 15 \pmod{21}$.

755

Check $k=7$:

$(49-7)/21=42/21=2$. $d_1=2$, $d_2=7-2=5$. $n=49$. In base 22, $49=2 \cdot 22 + 5=44 + 5=49$. Digits 2 and 5 sum to 7. Correct.

$k=15$:

$(225-15)/21=210/21=10$. $d_1=10$, $d_2=15-10=5$. $n=225$. In base 22, $225=10 \cdot 22 + 5=220 + 5=225$. Digits 10 and 5 sum to 15. Correct.

$k=21$:

$(441-21)/21=420/21=20$. $d_1=20$, $d_2=21-20=1$. $n=441$. In base 22, $441=20 \cdot 22 + 1=440 + 1=441$. Digits 20 and 1 sum to 21. Correct.

Thus, three beautiful numbers: 49, 225, 441. Still only three.

Next, try $b=10$. Already did that. Let's try $b=22$. Hmm, same as above.

等等，也许我们可以尝试一个 $\setminus(b-1)$ 是两个质数乘积的情况。我们来试一下 $\setminus(b=10)$ ，但是 $\setminus(9=3^2)$ 。或者， $\setminus(b=14)$ ： $\setminus(b-1=13)$ （质数）。那么在这种情况下，仅有两个解。

Alternatively, let's try $b=21$. Then $(b-1)=20=2^2 \cdot 5$.

Solutions to $k(k-1) \equiv 0 \pmod{20}$.

This requires that 4 divides $k(k-1)$ and 5 divides $k(k-1)$.

For 4 divides $k(k-1)$: $k \equiv 0$ or $1 \pmod{4}$.

For 5 divides $k(k-1)$: $k \equiv 0$ or $1 \pmod{5}$.

Using Chinese Remainder Theorem:

Combine solutions mod 4 and mod 5.

Possible combinations:

1. $k \equiv 0 \pmod{4}$ and $k \equiv 0 \pmod{5}$: $k \equiv 0 \pmod{20}$.

等等， $0 \pmod{4}$ 和 $0 \pmod{5}$ 的解是 $k \equiv 0 \pmod{20}$ 。

2. $k \equiv 0 \pmod{4}$ and $k \equiv 1 \pmod{5}$.

Solve $k=4m=5n+1$. Then $4m \equiv 1 \pmod{5} \Rightarrow 4m \equiv 1 \pmod{5} \Rightarrow m \equiv 4^{-1} \pmod{5}$. Since $4 \cdot 4 = 16 \equiv 1 \pmod{5}$, so inverse of 4 is 4.

Thus, $m \equiv 4 \pmod{5}$. Thus, $m=5p+4$. Then $k=4 \cdot (5p+4)=20p+16$. Therefore, $k \equiv 16 \pmod{20}$.

3. $k \equiv 1 \pmod{4}$ and $k \equiv 0 \pmod{5}$.

$k=4m+1=5n$. So $4m+1=5n$. Then $4m \equiv -1 \pmod{5} \Rightarrow 4m \equiv 4 \pmod{5} \Rightarrow m \equiv 1 \pmod{5}$. So $m=5p+1$. Then $k=4 \cdot (5p+1)+1=20p+5$. Therefore, $k \equiv 5 \pmod{20}$.

4. $k \equiv 1 \pmod{4}$ and $k \equiv 1 \pmod{5}$.

$k \equiv 1 \pmod{4}$ and $k \equiv 1 \pmod{5} \Rightarrow k \equiv 1 \pmod{20}$.

Thus, solutions are $k \equiv 0, 1, 5, 16 \pmod{20}$.

In the interval $[\text{ceil}(\sqrt{21}) \approx 5, 20]$, the numbers are 5, ..., 20.

Check $k \equiv 0, 1, 5, 16 \pmod{20}$:

$k=5 \equiv 5 \pmod{20}$

$k=16 \equiv 16 \pmod{20}$

$k=20 \equiv 0 \pmod{20}$

$k=1 \equiv 1 \pmod{20}$ (but $1 < 5$, excluded)

Thus, valid $k=5, 16, 20$.

Check $k=5$:

$(25 - 5)/20 = 20/20 = 1$. $d_1=1$, $d_2=5 - 1=4$. $n=25$. In base 21, $25 = 1 \cdot 21 + 4 = 25$. Digits 1 and 4 sum to 5. Correct.

$k=16$:

$(256 - 16)/20 = 240/20 = 12$. $d_1=12$, $d_2=16 - 12=4$. $n=256$. In base 21, $256 = 12 \cdot 21 + 4 = 252 + 4 = 256$. Digits 12 and 4 sum to 16. Correct.

$k=20$:

$(400 - 20)/20 = 380/20 = 19$. $d_1=19$, $d_2=20 - 19=1$. $n=400$. In base 21, $400 = 19 \cdot 21 + 1 = 399 + 1 = 400$. Digits 19 and 1 sum to 20. Correct.

Thus, three beautiful numbers: 25, 256, 400. Still three.

嗯，也许可以尝试 $\backslash(b = 32 \backslash)$ 。那么 $\backslash(b - 1 = 31 \backslash)$ ，这是一个质数。因此，在模 $\backslash(31 \backslash)$ 下，解的数量为 2。于是，在区间 $\backslash([\text{ceil}(\sqrt{32}) \approx 6, 31] \backslash)$ 中，解为 $\backslash(k = 31 \backslash)$ 。此时 $\backslash(n = 31^2 = 961 \backslash)$ 。在以 32 为基数下， $\backslash(961 = 30 \cdot 32 + 1 = 960 + 1 = 961 \backslash)$ 。其数字为 30 和 1，它们相加为 31。这是正确的。因此，仅有一个解。

Next, try $b=16$. Then $(b-1)=15=3 \cdot 5$.

Solutions to $k(k-1) \equiv 0 \pmod{15}$. So combinations of solutions mod 3 and mod 5.

mod 3: $k \equiv 0$ or $1 \pmod{3}$

mod 5: $k \equiv 0$ or $1 \pmod{5}$

Thus, four combinations:

1. $k \equiv 0 \pmod{3}$ and $k \equiv 0 \pmod{5}$: $k \equiv 0 \pmod{15}$.

2. $k \equiv 0 \pmod{3}$ and $k \equiv 1 \pmod{5}$: Solve $k=3m=5n+1$. $3m \equiv 1 \pmod{5}$. $3 \cdot 2 = 6 \equiv 1 \pmod{5}$, so $m \equiv 2 \pmod{5}$. $m=5p+2$. Then $k=3(5p+2)=15p+6$. Thus, $k \equiv 6 \pmod{15}$.

等等, $6 \bmod 3 = 0$, $6 \bmod 5 = 1$ 。正确。

3. $k \equiv 1 \pmod{3}$ and $k \equiv 0 \pmod{5}$: Solve $k=3m+1=5n$. $3m+1 \equiv 0 \pmod{5} \Rightarrow 3m \equiv -1 \equiv 4 \pmod{5}$. Multiply both sides by 2 (inverse of 3 mod 5 is 2, since $3 \cdot 2 = 6 \equiv 1 \pmod{5}$). So $m \equiv 8 \equiv 3 \pmod{5}$. $m=5p+3$. Then $k=3(5p+3)+1=15p+10$. Thus, $k \equiv 10 \pmod{15}$.

等等, $10 \bmod 3 = 1$, $10 \bmod 5 = 0$ 。正确。

4. $k \equiv 1 \pmod{3}$ and $k \equiv 1 \pmod{5}$: $k \equiv 1 \pmod{15}$.

Thus, solutions are $k \equiv 0, 1, 6, 10 \pmod{15}$.

In the interval $[\text{ceil}(\sqrt{16})=4, 15]$, numbers are 4, ..., 15.

Check $k \equiv 0, 1, 6, 10 \pmod{15}$:

$k=6, 10, 15 \equiv 6, 10, 0 \pmod{15}$.

$k=6$:

$(36-6)/15=30/15=2$. $d_1=2$, $d_2=6-2=4$. $n=36$. In base 16, $36=2 \cdot 16 + 4=32+4=36$. Digits 2 and 4 sum to 6. Correct.

等等, $6^2 = 36$ 。但是 6 等于 $\sqrt{36}$ 。没错。

$k=10$:

$(100-10)/15=90/15=6$. $d_1=6$, $d_2=10-6=4$. $n=100$. In base 16, $100=6 \cdot 16 + 4=96+4=100$. Digits 6 and 4 sum to 10.

Correct.

等等, $100=10^2$ 。所以 $\sqrt{100}=10$ 。对的。

$k=15$:

$(225-15)/15=210/15=14$. $d_1=14$, $d_2=15-14=1$. $n=225$. In base 16, $225=14 \cdot 16 + 1=224+1=225$. Digits 14 and 1 sum to 15. Correct.

Thus, three beautiful numbers: 36, 100, 225. So three solutions.

嗯, 看起来即使我们尝试更大的 $\lfloor b \rfloor$, 每种情况下只找到了 3 个解。也许可以尝试 $\lfloor b = 30 \rfloor$, 但我们已经发现只有一个解了。等等, 也许 $\lfloor b = 7 \rfloor$: $\lfloor b-1 = 6 = 2 \cdot 3 \rfloor$ 。

Solutions to $k(k-1) \equiv 0 \pmod{6}$. So combinations of solutions mod 2 and mod 3.

mod 2: $k \equiv 0$ or $1 \pmod{2}$

mod 3: $k \equiv 0$ or $1 \pmod{3}$

Thus, four combinations:

1. $k \equiv 0 \pmod{2}$ and $k \equiv 0 \pmod{3}$: $k \equiv 0 \pmod{6}$.
2. $k \equiv 0 \pmod{2}$ and $k \equiv 1 \pmod{3}$: $k \equiv 4 \pmod{6}$ (since $4 \equiv 0 \pmod{2}$, $4 \equiv 1 \pmod{3}$).
3. $k \equiv 1 \pmod{2}$ and $k \equiv 0 \pmod{3}$: $k \equiv 3 \pmod{6}$.
4. $k \equiv 1 \pmod{2}$ and $k \equiv 1 \pmod{3}$: $k \equiv 1 \pmod{6}$.

Thus, solutions are $k \equiv 0, 1, 3, 4 \pmod{6}$.

In the interval $[\text{ceil}(\sqrt{7}) \approx 3, 6]$, numbers are 3, 4, 5, 6.

Check $k \equiv 0, 1, 3, 4 \pmod{6}$:

$$k=3 \equiv 3 \pmod{6}$$

$$k=4 \equiv 4 \pmod{6}$$

$$k=6 \equiv 0 \pmod{6}$$

Check $k=3$:

$(9-3)/6=6/6=1$. $d_1=1$, $d_2=3-1=2$. $n=9$. In base 7, $9=1*7+2=7+2=9$. Digits 1 and 2 sum to 3. Correct.

$k=4$:

$(16-4)/6=12/6=2$. $d_1=2$, $d_2=4-2=2$. $n=16$. In base 7, $16=2*7+2=14+2=16$. Digits 2 and 2 sum to 4. Correct.

等等, $\sqrt{16}=4$ 。对的。

$k=6$:

$(36-6)/6=30/6=5$. $d_1=5$, $d_2=6-5=1$. $n=36$. In base 7, $36=5*7+1=35+1=36$. Digits 5 and 1 sum to 6. Correct.

Thus, three beautiful numbers: 9, 16, 36. So three solutions.

等等, 这表明当

$$(b-1)$$

是两个质数的乘积时, 解的数量是 4, 但在区间

$$[\text{ceil}(\sqrt{b}), b-1]$$

中, 可能会出现一些解被排除, 因此最终只剩下 3 个解。

Therefore, perhaps to get more solutions, we need $(b-1)$ to have more prime factors. Let's try a b where $(b-1)$ is the product of the first four primes: 2, 3, 5, 7. Then $(b-1)=235*7=210$, so $b=211$.

Then the number of solutions mod 210 is $2^4=16$. Therefore, in the interval $[\text{ceil}(\sqrt{211}) \approx 15, 210]$, the number of solutions would be 16 minus those less than 15. If none are less than 15, then 16 solutions. But let's check.

First, compute $\text{ceil}(\sqrt{211})$: $\sqrt{211} \approx 14.52$, so ceil is 15. Therefore, the interval is 15 to 210. The solutions mod 210 are $k \equiv 0, 1 \pmod{2, 3, 5, 7}$. Wait, no. Wait, the solutions are combinations of $k \equiv 0$ or $1 \pmod{\text{each prime factor}}$.

Since $210=235*7$, the number of solutions is $2^4=16$. Each solution corresponds to choosing for each prime p , $k \equiv 0$ or $1 \pmod{p}$. Then combine them using Chinese Remainder Theorem.

Each combination gives a unique solution mod 210. Therefore, there are 16 solutions mod 210. Therefore, in the interval $[0, 210)$, there are 16 solutions. Therefore, in the interval $[15, 210]$, the number of solutions is 16 minus the

number of solutions less than 15.

Check which of the 16 solutions are less than 15.

759

The 16 solutions mod 210 are:

For each prime factor 2, 3, 5, 7, choose $k \equiv 0$ or $1 \pmod{p}$. Then combine.

For example:

1. $k \equiv 0 \pmod{2,3,5,7}$: $k \equiv 0 \pmod{210}$. So $k=0 \pmod{210}$. In $[0, 210)$, this is 0. But 0 is excluded since $k \geq \lceil \sqrt{211} \rceil = 15$. So excluded.
2. $k \equiv 0 \pmod{2,3,5}$ and $k \equiv 1 \pmod{7}$: Solve $k \equiv 0 \pmod{(235)=30}$ and $k \equiv 1 \pmod{7}$. So $k=30m \equiv 1 \pmod{7}$. $30 \equiv 2 \pmod{7}$, so $2m \equiv 1 \pmod{7}$. Multiply both sides by inverse of 2 mod 7, which is 4 (since $2 \cdot 4 = 8 \equiv 1 \pmod{7}$). So $m \equiv 4 \pmod{7}$. Thus, $m=7n+4$. Then $k=30(7n+4)=210n+120$. Therefore, $k \equiv 120 \pmod{210}$.

Similarly, compute all 16 combinations. However, this would take time. But in any case, the solutions mod 210 are 16 numbers. Let's see how many of these are less than 15.

The 16 solutions mod 210 are:

1. 0 (excluded as before)
2. 1 (since $k \equiv 1 \pmod{2,3,5,7}$: $k \equiv 1 \pmod{210}$)
3. Numbers like 120 (as above), etc.

等等，也许我们可以枚举所有可能的组合：

For each prime factor, choose $k \equiv 0$ or $1 \pmod{p}$. So for primes 2, 3, 5, 7:

Each prime can be either 0 or 1, so 16 combinations.

Compute each combination:

1. All zeros: $k \equiv 0 \pmod{2,3,5,7} \Rightarrow k \equiv 0 \pmod{210}$. $k=0$ (excluded).
2. All ones: $k \equiv 1 \pmod{2,3,5,7} \Rightarrow k \equiv 1 \pmod{210}$. $k=1$ (excluded, <15).
3. $0 \pmod{2}$, $0 \pmod{3}$, $0 \pmod{5}$, $1 \pmod{7}$: $k \equiv 120 \pmod{210}$ as above.

Similarly, compute others. For example:

- $0 \pmod{2}, 0 \pmod{3}, 1 \pmod{5}, 1 \pmod{7}$: Compute $k \equiv 0 \pmod{6}$, $k \equiv 1 \pmod{5}$, $k \equiv 1 \pmod{7}$.

Compute $k \equiv 0 \pmod{6}$ and $k \equiv 1 \pmod{5}$ and $k \equiv 1 \pmod{7}$.

First, $k \equiv 0 \pmod{6}$ and $k \equiv 1 \pmod{5}$. Let $k=6m=5n+1$. Then $6m \equiv 1 \pmod{5} \Rightarrow m \equiv 1 \pmod{5}$. So $m=5p+1$. Then $k=6 \cdot (5p+1)=30p+6$. Now, $k \equiv 1 \pmod{7}$: $30p+6 \equiv 1 \pmod{7} \Rightarrow 30p \equiv -5 \pmod{7} \Rightarrow 30 \equiv 2 \pmod{7}$, so $2p \equiv 2 \pmod{7} \Rightarrow p \equiv 1 \pmod{7}$. Thus, $p=7q+1$. Then $k=30 \cdot (7q+1)+6=210q+36$. Therefore, $k \equiv 36 \pmod{210}$.

Similarly, compute all combinations. But this is tedious. However, the key point is that the 16 solutions mod 210 will be numbers from 0 to 209. Since we need $k \geq 15$, the number of solutions in $[15, 210]$ is 16 minus the number of

solutions less than 15.

So how many of the 16 solutions are less than 15?

The solutions include:

1. $k \equiv 0 \pmod{210}$
2. $k \equiv 1 \pmod{210}$
3. $k \equiv 1 \pmod{2,3,5,7}$

等等, 不对。等等, $k \equiv 1 \pmod{2,3,5,7}$: $k \equiv 1 \pmod{210}$, 因此 $k = 1$ 。

Other solutions:

For example, $k \equiv 0 \pmod{2,3,5}$ and $\equiv 1 \pmod{7}$: as above, $k \equiv 120 \pmod{210}$. 等等, 120。

Similarly, other combinations will result in numbers like 36, etc.

So likely, the solutions less than 15 are only 0 and 1, which are excluded. Therefore, all 16 solutions are ≥ 15 ? But 16 solutions in $[0, 210)$ are 0, 1, and 14 others between 1 and 209. But if 0 and 1 are excluded, then 14 solutions in $[1, 210)$. But $\text{ceil}(\sqrt{211}) = 15$, so we need $k \geq 15$. Therefore, the number of solutions is 14 (since 0 and 1 are excluded, but some solutions might be between 2 and 14). Wait, but if all solutions are ≥ 15 , then we have 14 solutions. However, let's check.

For example, $k \equiv 1 \pmod{210}$: 1 (excluded). $k \equiv 120 \pmod{210}$: 120. $k \equiv 36 \pmod{210}$: 36. Similarly, other combinations will result in numbers ≥ 36 , 120, etc. So likely, all solutions except 0 and 1 are ≥ 36 , which is ≥ 15 . Therefore, the number of solutions in $[15, 210]$ is $16 - 2 = 14$. Therefore, 14 beautiful numbers. But wait, 14 is more than ten. Therefore, $b = 211$ would have 14 beautiful numbers. Therefore, the minimal b is 211. But the problem asks for the least integer $b \geq 2$ for which there are more than ten b -beautiful integers. Therefore, 211. But let's confirm.

But before confirming, let's check if there are smaller b with more solutions. For example, if $(b-1)$ is a product of three primes, say $2 \cdot 3 \cdot 5 = 30$, so $b = 31$. Then the number of solutions mod 30 is $2^3 = 8$. Therefore, in the interval $[\text{ceil}(\sqrt{31}) \approx 6, 30]$, the number of solutions would be 8 minus those less than 6. Let's compute.

Solutions to $k(k-1) \equiv 0 \pmod{30}$. Which requires $k \equiv 0$ or $1 \pmod{2,3,5}$.

Using Chinese Remainder Theorem, the solutions are combinations of:

- $k \equiv 0$ or $1 \pmod{2}$
- $k \equiv 0$ or $1 \pmod{3}$

等等, 3 个素数: 2、3、5。

So the combinations are:

1. $k \equiv 0 \pmod{2,3,5}$: $k \equiv 0 \pmod{30}$
2. $k \equiv 0 \pmod{2,3}$ and $\equiv 1 \pmod{5}$: $k \equiv 0 \pmod{6}$ and $\equiv 1 \pmod{5}$. Solve $k = 6m \equiv 1 \pmod{5} \Rightarrow 6m \equiv 1 \pmod{5} \Rightarrow m \equiv 1 \pmod{5}$. So $m = 5n + 1$. Then $k = 6 \cdot (5n + 1) = 30n + 6 \equiv 6 \pmod{30}$.

3. $k \equiv 0 \pmod{2,5}$ and $\equiv 1 \pmod{3}$: $k \equiv 0 \pmod{10}$ and $\equiv 1 \pmod{3}$. Solve $k=10m \equiv 1 \pmod{3} \Rightarrow 10m \equiv 1 \pmod{3} \Rightarrow m \equiv 1 \pmod{3}$.

So $m=3n+1$. Then $k=10*(3n+1)=30n+10 \equiv 10 \pmod{30}$.

761

等等, $10 \pmod{3} = 1$ 。因此, $10 \equiv 1 \pmod{3}$ 。所以, 这是正确的。

4. $k \equiv 0 \pmod{2}$ and $\equiv 1 \pmod{3,5}$: $k \equiv 0 \pmod{2}$ and $\equiv 1 \pmod{3,5}$. Solve $k \equiv 1 \pmod{3,5}$ and $\equiv 0 \pmod{2}$. Find $k \equiv 1 \pmod{15}$ and $\equiv 0 \pmod{2}$. Solve $k=15m+1 \equiv 0 \pmod{2} \Rightarrow 15m+1 \equiv m+1 \pmod{2} \equiv 0 \Rightarrow m \equiv 1 \pmod{2}$. So $m=2n+1$. Then $k=15*(2n+1)+1=30n+16 \equiv 16 \pmod{30}$.

Similarly, other combinations:

5. $k \equiv 0 \pmod{3,5}$ and $\equiv 1 \pmod{2}$: $k \equiv 0 \pmod{15}$ and $\equiv 1 \pmod{2}$: $k=15m \equiv 1 \pmod{2} \Rightarrow 15m \equiv 1 \pmod{2} \Rightarrow m \equiv 1 \pmod{2}$. So $m=2n+1$. Then $k=15(2n+1)=30n+15 \equiv 15 \pmod{30}$.

6. $k \equiv 0 \pmod{3}$ and $\equiv 1 \pmod{2,5}$: $k \equiv 0 \pmod{3}$ and $\equiv 1 \pmod{2,5}$. Solve $k \equiv 1 \pmod{2,5}$ and $\equiv 0 \pmod{3}$. $k \equiv 1 \pmod{10}$ and $\equiv 0 \pmod{3}$. Solve $k=10m+1 \equiv 0 \pmod{3} \Rightarrow 10m+1 \equiv m+1 \pmod{3} \Rightarrow m \equiv 2 \pmod{3}$. So $m=3n+2$. Then $k=10*(3n+2)+1=30n+21 \equiv 21 \pmod{30}$.

7. $k \equiv 0 \pmod{5}$ and $\equiv 1 \pmod{2,3}$: $k \equiv 0 \pmod{5}$ and $\equiv 1 \pmod{2,3}$. Solve $k \equiv 0 \pmod{5}$ and $\equiv 1 \pmod{6}$. $k=5m \equiv 1 \pmod{6}$. $5m \equiv 1 \pmod{6}$. 由于 $5 \equiv -1 \pmod{6}$, 因此方程变为 $-m \equiv 1 \pmod{6}$, 即 $m \equiv 5 \pmod{6}$ 。所以, $m=6n+5$ 。接着, $k=5*(6n+5)=30n+25 \equiv 25 \pmod{30}$ 。

8. $k \equiv 1 \pmod{2,3,5}$: $k \equiv 1 \pmod{30}$.

Thus, the 8 solutions mod 30 are 0, 6, 10, 15, 16, 21, 25, 30. 等等, 0 和 30 是相同的。所以这些解为: 0, 6, 10, 15, 16, 21, 25, 1 (因为当 $k=1$ 时, 模 30 下的结果是 1)。等等, 不对。等等, 第 8 个解是 $(k \equiv 1 \pmod{30})$ 。所以这些解是 $(k \equiv 0, 1, 6, 10, 15, 16, 21, 25 \pmod{30})$ 。

But in the interval $[6, 30]$, the solutions are 6, 10, 15, 16, 21, 25, 30 $\equiv 0 \pmod{30}$ (excluded), and $1 \equiv 1 \pmod{30}$ (excluded). So valid $k=6, 10, 15, 16, 21, 25$.

Check each:

$k=6$:

$(36-6)/30=30/30=1$. $d_1=1$, $d_2=6-1=5$. $n=36$. In base 31, $36=1*31+5=36$. Digits 1 and 5 sum to 6. Correct.

$k=10$:

$(100-10)/30=90/30=3$. $d_1=3$, $d_2=10-3=7$. $n=100$. In base 31, $100=3*31+7=93+7=100$. Digits 3 and 7 sum to 10.

Correct.

$k=15$:

$(225-15)/30=210/30=7$. $d_1=7$, $d_2=15-7=8$. $n=225$. In base 31, $225=7*31+8=217+8=225$. Digits 7 and 8 sum to 15.

Correct.

$k=16$:

$(256-16)/30=240/30=8$. $d_1=8$, $d_2=16-8=8$. $n=256$. In base 31, $256=8*31+8=248+8=256$. Digits 8 and 8 sum to 16.

Correct.

k=21:

$(441 - 21)/30 = 420/30 = 14$. $d_1 = 14$, $d_2 = 21 - 14 = 7$. $n = 441$. In base 31, $441 = 14 \cdot 31 + 7 = 434 + 7 = 441$. Digits 14 and 7 sum to 21. Correct.

等等, $21^2 = 441$ 。是的。

k=25:

$(625 - 25)/30 = 600/30 = 20$. $d_1 = 20$, $d_2 = 25 - 20 = 5$. $n = 625$. In base 31, $625 = 20 \cdot 31 + 5 = 620 + 5 = 625$. Digits 20 and 5 sum to 25. Correct.

Thus, six beautiful numbers: 36, 100, 225, 256, 441, 625. So six solutions. Which is more than the previous examples, but still less than ten. Therefore, $b = 31$ has six beautiful numbers.

Therefore, moving to $b = 211$ would give 14, which is more than ten. But before that, check if there is a smaller b . For example, take $(b - 1) = 235 = 30$, leading to $b = 31$ with six solutions. If we take $(b - 1) = 235 \cdot 7 = 210$, leading to $b = 211$ with 14 solutions. Alternatively, maybe a smaller $(b - 1)$ with three prime factors, but with overlapping solutions.

Alternatively, take $(b - 1) = 237 = 42$, so $b = 43$. Then number of solutions mod 42 is $2^3 = 8$. Therefore, in the interval $[\text{ceil}(\sqrt{43}) \approx 7, 42]$, the number of solutions would be 8 minus those less than 7.

Compute solutions mod 42:

Solutions are combinations of $k \equiv 0$ or $1 \pmod{2, 3, 7}$.

Therefore, 8 solutions:

1. $k \equiv 0 \pmod{2, 3, 7}$: $k \equiv 0 \pmod{42}$.

等待:

1. $0 \pmod{2, 3, 7}$: $0 \pmod{42}$.

等等, 这样会得到以下 8 种组合:

For each prime, choose 0 or 1:

- $0, 0, 0 \pmod{42}$.

等等, 这会引发一些疑惑。我们来枚举所有组合吧:

1. $0 \pmod{2}, 0 \pmod{3}, 0 \pmod{7}$: $0 \pmod{42}$.

等等, 这里可能需要同时满足以下条件: $k \pmod{2} = 0$, $k \pmod{3} = 0$, $k \pmod{7} = 0$ 。因此, $k \equiv 0 \pmod{42}$ 。

2. $0 \pmod{2}, 0 \pmod{3}, 1 \pmod{7}$: $0 \pmod{6}, 1 \pmod{7}$. Solve $k = 6m \equiv 1 \pmod{7}$. $6m \equiv 1 \pmod{7}$. $6 \equiv -1 \pmod{7}$, so $-m \equiv 1 \pmod{7} \Rightarrow m \equiv -1 \equiv 6 \pmod{7}$. So $m = 7n + 6$. Then $k = 6 \cdot (7n + 6) = 42n + 36 \equiv 36 \pmod{42}$.

3. $0 \pmod{2}, 1 \pmod{3}, 0 \pmod{7}$: $0 \pmod{2}, 0 \pmod{7}$. So $k \equiv 0 \pmod{14}$. Also, $1 \pmod{3}$. Solve $k = 14m \equiv 1 \pmod{3}$. $14 \equiv 2 \pmod{3}$, so $2m \equiv 1 \pmod{3} \Rightarrow m \equiv 2 \pmod{3}$. So $m = 3n + 2$. Then $k = 14 \cdot (3n + 2) = 42n + 28 \equiv 28 \pmod{42}$.

等等, $28 \pmod{3} = 1$, 因为 $28 \div 3 = 9 \dots 1$ 。所以, $28 \equiv 1 \pmod{3}$ 。这是正确的。

4. $0 \pmod{2}, 1 \pmod{3}, 1 \pmod{7}$: $0 \pmod{2}, 1 \pmod{3}, 1 \pmod{7}$. Solve $k \equiv 1 \pmod{3}, 1 \pmod{7}$, and $0 \pmod{2}$. $k \equiv 1 \pmod{21}$ and $0 \pmod{2}$. Since 21 is odd, $k \equiv 1 \pmod{21}$ and $0 \pmod{2}$. But $1 \equiv 1 \pmod{21}$ and $0 \pmod{2}$, which is impossible. Therefore, no solution here.

763

等等，这有点让人困惑。我们来仔细看看。

$k \equiv 1 \pmod{3}$ and $1 \pmod{7}$: $k \equiv 1 \pmod{21}$.

$k \equiv 0 \pmod{2}$.

So $k \equiv 1 \pmod{21}$ and $0 \pmod{2}$. So k must be even and $\equiv 1 \pmod{21}$. But $1 \pmod{21}$ is odd. Therefore, no solution.

Therefore, this combination is invalid.

So combination 4 is invalid.

5. $1 \pmod{2}, 0 \pmod{3}, 0 \pmod{7}$: $1 \pmod{2}, 0 \pmod{21}$. Similarly, impossible. Because $0 \pmod{21}$ is even, but $1 \pmod{2}$ is odd. Therefore, no solution.

6. $1 \pmod{2}, 0 \pmod{3}, 1 \pmod{7}$: $1 \pmod{2}, 0 \pmod{3}, 1 \pmod{7}$. Solve $k \equiv 1 \pmod{2}, 0 \pmod{3}, 1 \pmod{7}$.

$k \equiv 1 \pmod{2}$ and $0 \pmod{3}$. Let $k = 3m \equiv 1 \pmod{2}$. $3m \equiv 1 \pmod{2} \Rightarrow m \equiv 1 \pmod{2}$. So $m = 2n + 1$. Then $k = 3 \cdot (2n + 1) = 6n + 3 \equiv 3 \pmod{6}$.

Also, $k \equiv 1 \pmod{7}$. So $6n + 3 \equiv 1 \pmod{7} \Rightarrow 6n \equiv -2 \equiv 5 \pmod{7}$. Multiply both sides by inverse of 6 mod 7, which is 6 (since $6 \cdot 6 = 36 \equiv 1 \pmod{7}$). So $n \equiv 5 \cdot 6 = 30 \equiv 30 - 4 \cdot 7 = 2 \pmod{7}$. So $n = 7p + 2$. Then $k = 6(7p + 2) + 3 = 42p + 15$. Therefore, $k \equiv 15 \pmod{42}$.

等等， $15 \pmod{2} = 1$ ，这是正确的，因为 15 是奇数。

$k = 15 \equiv 0 \pmod{3}, 3 \cdot 5$.

$k = 15 \equiv 1 \pmod{7}$, since $15 = 2 \cdot 7 + 1$.

Thus, valid.

7. $1 \pmod{2}, 1 \pmod{3}, 0 \pmod{7}$: $1 \pmod{2}, 1 \pmod{3}, 0 \pmod{7}$. Solve $k \equiv 1 \pmod{2}, 1 \pmod{3}, 0 \pmod{7}$.

$k \equiv 1 \pmod{2}$ and $1 \pmod{3}$. Then $k \equiv 4 \pmod{6}$.

Also, $k \equiv 0 \pmod{7}$. Solve $k = 7m \equiv 4 \pmod{6}$. $7m \equiv m \equiv 4 \pmod{6}$. So $m \equiv 4 \pmod{6}$. $m = 6n + 4$. Then $k = 7 \cdot (6n + 4) = 42n + 28 \equiv 28 \pmod{42}$.

Check $k = 28$:

$28 \pmod{2} = 0 \equiv 1 \pmod{2}$? No. Wait, 28 is even, but we need $k \equiv 1 \pmod{2}$. So invalid. Therefore, no solution here.

等等，这里有问题。我们重新解决一下：

$k \equiv 1 \pmod{2}, 1 \pmod{3}, 0 \pmod{7}$.

So $k = 7m$, and $k \equiv 1 \pmod{2}$ and $1 \pmod{3}$.

$7m \equiv 1 \pmod{2} \Rightarrow m \equiv 1 \pmod{2}$.

$$7m \equiv 1 \pmod{3} \Rightarrow 7m \equiv m \equiv 1 \pmod{3}.$$

Therefore, $m \equiv 1 \pmod{2}$ and $m \equiv 1 \pmod{3}$. So $m \equiv 1 \pmod{\text{lcm}(2,3)=6}$. Thus, $m=6n+1$. Then $k=7*(6n+1)=42n+7 \equiv 7 \pmod{42}$.

Therefore, $k \equiv 7 \pmod{42}$. Check:

$7 \pmod{2}=1$, $7 \pmod{3}=1$, $7 \pmod{7}=0$. Correct. So $k=7$. But 7 is less than $\text{ceil}(\sqrt{43}) \approx 6.93$, so ceil is 7. So $k=7$ is included. Therefore, valid.

8. $1 \pmod{2}, 1 \pmod{3}, 1 \pmod{7}: 1 \pmod{2}, 1 \pmod{3}, 1 \pmod{7}$. Solve $k \equiv 1 \pmod{2}, 1 \pmod{3}, 1 \pmod{7}$. Which is $k \equiv 1 \pmod{\text{lcm}(2,3,7)=42}$. So $k \equiv 1 \pmod{42}$.

Thus, solutions are $k \equiv 0, 7, 15, 21, 28, 35, 42 \equiv 0 \pmod{42}$, but $k=7, 15, 21, 28, 35, 1$.

But in the interval $[7, 42]$, the solutions are $7, 15, 21, 28, 35, 42$. However, 42 is equivalent to $0 \pmod{42}$, which is excluded. So valid $k=7, 15, 21, 28, 35$.

Check each:

$k=7$:

$(49-7)/42=42/42=1$. $d_1=1$, $d_2=7-1=6$. $n=49$. In base 43, $49=1*43+6=49$. Digits 1 and 6 sum to 7. Correct.

$k=15$:

$(225-15)/42=210/42=5$. $d_1=5$, $d_2=15-5=10$. $n=225$. In base 43, $225=5*43+10=215+10=225$. Digits 5 and 10 sum to 15. Correct.

等等, $5+10=15$ 。没错。

$k=21$:

$(441-21)/42=420/42=10$. $d_1=10$, $d_2=21-10=11$. $n=441$. In base 43, $441=10*43+11=430+11=441$. Digits 10 and 11 sum to 21. Correct.

$k=28$:

$(784-28)/42=756/42=18$. $d_1=18$, $d_2=28-18=10$. $n=784$. In base 43, $784=18*43+10=774+10=784$. Digits 18 and 10 sum to 28. Correct.

$k=35$:

$(1225-35)/42=1190/42=28.5$. Not integer. 等等, 1190 除以 42 是 28.333...。等等, 这不可能。

等等, $1225-35=1190$ 。1190 除以 42: $42 \times 28 = 1176$, $1190-1176=14$ 。所以

$1190=42 \times 28+14$ 。因此, $1190/42=28+14/42=28+1/3$ 。这不是整数。因此, $k=35$ 不是有效解。等等, 但在模 42 下, $k=35$ 是一个解。然而, $(k^2-k)/42$ 并不是整数。这是因为虽然 42 可以整除 $k(k-1)$, 但 $k=35$ 时, $k \equiv 35 \pmod{42}$, $k-1=34 \pmod{42}$ 。35 和 34 在模 42 下分别是 35 和 34, 但它们的乘积是 $35 \times 34 = 1190$ 。1190 模 42 等于 14, 因为 $1190 \div 42 = 28$ 余 14。

因此, $1190 \equiv 14 \pmod{42}$ 。也就是说, 42 可以整除 1190 除以 42 的结果, 但 $1190/42$ 并不是整数。这表明虽然满足模 42 的条件, 但 $(k^2 - k)/42$ 并不一定是整数。等等, 但 $k(k-1)$ 必须能被 42 整除。然而, $35 \cdot 34 = 1190$, 而 1190 并不能被 42 整除。等等, 42 是 $2 \cdot 3 \cdot 7$ 。接下来检查:

1190 divided by 2: $1190/2=595$. Divided by 3: $595/3 \approx 198.333$, not integer. Divided by 7: $1190/7=170$. So $1190=25717$. Therefore, $42=237$ divides 1190? $42=237$, but $1190=25717$. The 3 is missing. Therefore, 42 does not divide 1190. Therefore, $k=35$ is not a valid solution. Wait, but according to the previous analysis, $k=35$ is a solution mod 42. But actually, $k=35 \equiv -7 \pmod{42}$, but $k(k-1)=3534=1190$. $1190 \bmod 42 = 1190 - 42 \cdot 28 = 1190 - 1176 = 14$. Therefore, $14 \equiv 0 \pmod{14}$, but 42 does not divide 1190. Therefore, there is a contradiction here. Therefore, our earlier assumption that combining the modular solutions would lead to $k(k-1) \equiv 0 \pmod{42}$ is incorrect. Because even though $k \equiv 0$ or $1 \pmod{p}$ for each prime, the product $k(k-1)$ might not be divisible by the product of the primes if the factors are distributed across the primes.

等等, 这到底是怎么回事? 让我们来仔细思考一下。假设我们需要满足 $k(k-1) \equiv 0 \pmod{42}$ 。由于 $42 = 2 \cdot 3 \cdot 7$, 因此 $2 \cdot 3 \cdot 7$ 都必须整除 $k(k-1)$ 。因为 k 和 $k-1$ 是互质的, 所以每个素数必须整除 k 或 $k-1$ 。也就是说, 对于每个素数 p (这里为 $2, 3, 7$), 必须满足 p 整除 k 或 p 整除 $k-1$ 。因此, 对于每个素数, 我们有 $k \equiv 0 \pmod{p}$ 或 $k \equiv 1 \pmod{p}$ 。因此, 该同余方程的解是所有可能的组合, 即每个素数取 0 或 1 的情况。对于三个素数, 共有 $2^3 = 8$ 种组合。

然而, 当我们取 $k = 35$ 时, 它在模 2、3、7 下分别满足 $k \equiv 1 \pmod{2}$ 、 $k \equiv 2 \pmod{3}$ 、 $k \equiv 0 \pmod{7}$ 。等等, 不对。等等, $35 \bmod 2 = 1$, $35 \bmod 3 = 2$, $35 \bmod 7 = 0$ 。再等等, 根据 3 来说, $35 \equiv 2 \pmod{3}$, 这并不是 0 或 $1 \pmod{3}$ 。因此, $k = 35$ 并不满足 $k \equiv 0$ 或 $1 \pmod{3}$ 的条件。等等, 但在之前的分析中, 我们考虑的是 $k \equiv 0$ 或 $1 \pmod{2}$ 、 0 或 $1 \pmod{3}$ 、 0 或 $1 \pmod{7}$ 的组合。对于 $k = 35$, 我们有 $k \equiv 1 \pmod{2}$ 、 $k \equiv 2 \pmod{3}$ 、 $k \equiv 0 \pmod{7}$ 。但这并不是一个合法的组合。等等, 这说明哪里出错了?

等等, 不对。当考虑模 42 的解时, 解的集合是所有满足以下条件的 k : 对每个素因子 p , 都有 $k \equiv 0 \pmod{p}$ 或 $k \equiv 1 \pmod{p}$ 。对于 $p = 2$, $k \equiv 1 \pmod{2}$; 对于 $p = 3$, $k \equiv 2 \pmod{3}$; 对于 $p = 7$, $k \equiv 0 \pmod{7}$ 。然而, $2 \pmod{3}$ 既不是 0 也不是 $1 \pmod{3}$, 所以 $k = 35$ 并不满足条件。因此, 之前的分析中存在错误。

在我们之前的分析中, 我们寻找了满足以下条件的解:

对于每个素因子 p , 都有 $k \equiv 0 \pmod{p}$ 或 $1 \pmod{p}$ 。

因此, $k = 35$ 对于 $p = 7$ 满足 $0 \pmod{7}$, 但对 $p = 3$ 满足 $2 \pmod{3}$, 这不符合条件。所以, $k = 35$ 并不算是一个解。

然而, 在我们列举的 8 个解中, $k = 35$ 被错误地包含进去了。让我们重新审视一下之前的解法。

We had combinations:

1. $0,0,0:0 \bmod 42$.
2. $0,0,1:36 \bmod 42$.
3. $0,1,0:28 \bmod 42$.
4. Invalid.

等等，第四种组合是无效的。

5. Invalid.

$6.15 \bmod 42$.

$7.7 \bmod 42$.

等等，第七种是 $k \equiv 7 \pmod{42}$ 。

$8.1 \bmod 42$.

But in the interval $[7,42]$, the valid solutions are 7,15,21,28,35. But 35 is invalid as above. Similarly, 21:

$k=21$:

Check $k=21$. For $p=2:21 \equiv 1 \bmod 2$. For $p=3:21 \equiv 0 \bmod 3$. For $p=7:21 \equiv 0 \bmod 7$. Therefore, for $p=3$ and $p=7, 21 \equiv 0 \bmod 3$ and $0 \bmod 7$, which is allowed (since 0 is allowed). So $k=21$ is a valid solution. Because for each prime, $k \equiv 0 \bmod p$ or $1 \bmod p$. For $p=2, 21 \equiv 1$; for $p=3, 21 \equiv 0$; for $p=7, 21 \equiv 0$. Therefore, valid.

But $k=21$:

$(441 - 21)/42 = 420/42 = 10$. Which is integer. So valid. Therefore, $n = 441 = 21^2$. In base 43, $441 = 10 \cdot 43 + 11 = 430 + 11 = 441$. Digits 10 and 11 sum to 21. Correct.

Similarly, $k=28$:

$k=28$:

For $p=2:28 \equiv 0 \bmod 2$.

$p=3:28 \equiv 1 \bmod 3$.

$p=7:28 \equiv 0 \bmod 7$.

Therefore, for $p=2, 28 \equiv 0$; $p=3, 28 \equiv 1$; $p=7, 28 \equiv 0$. Therefore, valid.

Thus, $k=28$ is valid. But earlier, we thought $k=28$ would lead to non-integer, but actually:

$(784 - 28)/42 = 756/42 = 18$. Which is integer. Therefore, $d_1 = 18$, $d_2 = 28 - 18 = 10$. $n = 784 = 28^2$. In base 43, $784 = 18 \cdot 43 + 10 = 774 + 10 = 784$. Digits 18 and 10 sum to 28. Correct.

Therefore, the valid k 's in $[7,42]$ are 7,15,21,28. $k=35$ is invalid. So four solutions. Therefore, in base 43, there are four beautiful numbers. Which is more than previous examples but still less than ten.

Therefore, moving to $b=211$ with 14 solutions is the first instance where the number exceeds ten. Therefore, the minimal b is 211.

But let's check another example. Let's take $(b-1)=235*7=210$, $b=211$. Then the number of solutions is 16. As before, in the interval $[15,210]$, assuming all solutions are ≥ 15 , we have 16 solutions. Each solution corresponds to a combination of $k \equiv 0$ or $1 \pmod{2,3,5,7}$. For example:

$k \equiv 0 \pmod{2,3,5,7}$: $k=0 \pmod{210}$. Excluded.

$k \equiv 1 \pmod{2,3,5,7}$: $k=1 \pmod{210}$. Excluded.

Then, other combinations like:

$k \equiv 0 \pmod{2,3,5}$ and $1 \pmod{7}$: $k=120 \pmod{210}$ as before.

Similarly, each combination gives a unique k in $[0,210)$. The solutions are:

$0, 1, 30, 36, 70, 84, 90, 120, 126, 150, 156, 180, 186, 210 \equiv 0$. So the solutions are $0, 1, 30, 36, 70, 84, 90, 120, 126, 150, 156, 180, 186$. But 0 and 1 are excluded. Now check which of these are ≥ 15 .

Starting from 30 upwards:

$30, 36, 70, 84, 90, 120, 126, 150, 156, 180, 186$. So eleven solutions. But wait, $120, 126, 150, 156, 180, 186$. Let's count:

From 30 to 186, how many solutions? Let's list them:

$30, 36, 70, 84, 90, 120, 126, 150, 156, 180, 186$. That's eleven numbers. Wait, but earlier I thought there were 16 solutions.

But perhaps some solutions are less than 15. For example, $k=6 \pmod{70}$? Wait, no. Let me recount.

The solutions $\pmod{210}$ are:

For each prime, choose 0 or 1. So 16 combinations. The solutions are:

1. $0 \pmod{210}$.

2. $1 \pmod{210}$.

3. $0 \pmod{2,3,5}$ and $1 \pmod{7}$: $30*1=30 \pmod{210}$.

等等，不对：

Wait, for $k \equiv 0 \pmod{2,3,5}$ and $1 \pmod{7}$. As before, $k=30m \equiv 1 \pmod{7}$. 30 除以 7 的余数为 2，因此 $30m \equiv 2m \pmod{7}$ 。求解 $2m \equiv 1 \pmod{7}$ 。求解 m ：

$2m \equiv 1 \pmod{7}$ ，其解为 $m \equiv 4 \pmod{7}$ ，因为 $2 \times 4 = 8 \equiv 1 \pmod{7}$ 。

于是， $m = 7n + 4$ ，代入可得：

$k = 30 \times (7n + 4) = 210n + 120$ 。

因此， $k \equiv 120 \pmod{210}$ 。

Similarly, other combinations:

$k \equiv 0 \pmod{2,3,7}$ and $1 \pmod{5}$: Solve $k \equiv 0 \pmod{42}$ and $1 \pmod{5}$. $k=42m \equiv 1 \pmod{5}$. $42 \equiv 2 \pmod{5}$ ，因此 $2m \equiv 1 \pmod{5}$ 。解得 $m \equiv 3 \pmod{5}$ 。所以 $m = 5n + 3$ ，代入得：

$k = 42 \times (5n + 3) = 210n + 126 \equiv 126 \pmod{210}$ 。

Similarly, others:

$k \equiv 0 \pmod{2,5,7}$ and $1 \pmod{3}$: Solve $k \equiv 0 \pmod{70}$ and $1 \pmod{3}$. $70m \equiv 1 \pmod{3} \Rightarrow 70 \equiv 1 \pmod{3}$, so $1m \equiv 1 \pmod{3} \Rightarrow m \equiv 1 \pmod{3}$. $m = 3n + 1$. Then $k = 70(3n + 1) = 210n + 70 \equiv 70 \pmod{210}$.

Thus, solutions like 70, 126, 120, etc.

Listing all 16 solutions mod 210:

0, 1, 30, 36, 70, 84, 90, 120, 126, 150, 156, 180, 186, 210 (which is 0), etc. Wait, actually, the solutions are all numbers where for each prime, they are $\equiv 0$ or 1. Therefore, the numbers are:

For each prime factor 2, 3, 5, 7:

- If we choose 0 for all, we get 0.
- If we choose 1 for all, we get 1.
- Other combinations:

For example:

- $0 \pmod{2,3,5}, 1 \pmod{7}$: 120
- $0 \pmod{2,3,7}, 1 \pmod{5}$: 126

等等，我需要重新生成这16个解。

But to save time, let's note that in the interval $[15, 210]$, the solutions are from 30 up to 186, which are eleven numbers. Wait, but earlier we thought there were 16 solutions. Therefore, perhaps my previous assumption that all solutions are ≥ 15 is incorrect. Let's check some solutions:

For example, $k \equiv 0 \pmod{2,3,5}, 1 \pmod{7}$: 120

$k \equiv 0 \pmod{2,3,7}, 1 \pmod{5}$: 126

$k \equiv 0 \pmod{2,5,7}, 1 \pmod{3}$: 70

$k \equiv 0 \pmod{3,5,7}, 1 \pmod{2}$: 105 (Wait, $105 \pmod{2} = 1$, but 105 is odd. Let's check:

If we choose $k \equiv 1 \pmod{2}, 0 \pmod{3,5,7}$. Then $k \equiv 0 \pmod{105}$ and $1 \pmod{2}$. Solve $k = 105m \equiv 1 \pmod{2}$. $105 \equiv 1 \pmod{2}$, so $m \equiv 1 \pmod{2}$. $m = 2n + 1$. Then $k = 105(2n + 1) = 210n + 105 \equiv 105 \pmod{210}$. So $k = 105$. Check 105:

105 in base 211 is a two-digit number. Let's check if it's b-beautiful.

$n = 105^2 = 11025$. In base 211, 11025 divided by 211:

$211 \cdot 52 = 10972$, $11025 - 10972 = 53$. So $11025 = 52 \cdot 211 + 53$. Digits 52 and 53. Their sum is $52 + 53 = 105$. $\sqrt{11025} = 105$.

Correct.

Therefore, $k = 105$ is a valid solution. Similarly, $k = 105$ is in the interval $[15, 210]$.

Similarly, $k = 165$:

Another combination. For example, $k \equiv 1 \pmod{2}, 0 \pmod{3,5,7}$. Then $k \equiv 0 \pmod{105}$ and $1 \pmod{2}$. As above, $k=105m+105$. Wait, no.

769

等等, $105 \pmod{2} = 1$, 所以 $k=105 \equiv 1 \pmod{2}$. 但如果选择条件是 $k \equiv 0 \pmod{3,5,7}$ 和 $k \equiv 1 \pmod{2}$, 那么 $k=105$ 是符合条件的。

Similarly, $k=105$ is a solution. So in total, the solutions in $[15,210]$ are:

30,36,70,84,90,105,120,126,150,156,180,186. Let's count these:

From 30 to 186, how many numbers?

30,36 (difference 6), 70 (difference 34), 84 (difference 14), 90 (difference 6), 105 (difference 15), 120 (difference 15), 126 (difference 6), 150 (difference 24), 156 (difference 6), 180 (difference 24), 186 (difference 6). Total of 11 numbers. Wait, but this is only 11. But we need 16 solutions. Where are the other 5?

等等, 可能还有其他的解。例如, $k \equiv 0 \pmod{2,3}, 1 \pmod{5,7}$ 。

Let's compute:

$k \equiv 0 \pmod{6}$ and $1 \pmod{5}, 1 \pmod{7}$.

Solve $k \equiv 0 \pmod{6}$ and $k \equiv 1 \pmod{5}, 1 \pmod{7}$.

First, solve $k \equiv 0 \pmod{6}$ and $k \equiv 1 \pmod{5}$.

$k=6m \equiv 1 \pmod{5} \Rightarrow 6m \equiv 1 \pmod{5} \Rightarrow m \equiv 1 \pmod{5}$. $m=5n+1$. Then $k=6*(5n+1)=30n+6 \equiv 6 \pmod{30}$.

Then solve $k \equiv 6 \pmod{30}$ and $k \equiv 1 \pmod{7}$.

$k=30n+6 \equiv 1 \pmod{7} \Rightarrow 30n+6 \equiv 1 \pmod{7} \Rightarrow 30 \equiv 2 \pmod{7}$, so $2n+6 \equiv 1 \pmod{7} \Rightarrow 2n \equiv -5 \equiv 2 \pmod{7} \Rightarrow n \equiv 1 \pmod{7}$. $n=7p+1$.

Then $k=30*(7p+1)+6=210p+36$. Therefore, $k \equiv 36 \pmod{210}$.

Therefore, $k=36$. Check $k=36$:

$(36^2 - 36)/210 = (1260)/210 = 6$. $d_1=6$, $d_2=36-6=30$. $n=1296$. In base 211, $1296=6*211+30=1266+30=1296$. Digits 6 and 30 sum to 36. Correct.

Thus, $k=36$ is a solution. So this was missing earlier. Therefore, the solutions include numbers like 36, 70, 105, etc.

Therefore, the total number of solutions in $[15,210]$ is 16 minus the solutions less than 15. The solutions less than 15 are 0 and 1. Therefore, $16-2=14$ solutions. Therefore, 14 solutions. Each of these k 's from 30 to 210 (but actually, some are less than 15? Wait, no. $k=36$ is in the interval $[15,210]$. Similarly, $k=30,36,70,84,90,105,120,126,150,156,180,186$. That's twelve numbers. Wait, but $16-2=14$. So where are the other two?

Wait, perhaps $k=165$ and $k=168$.

等等, 让我检查一下解法。

Another combination is $k \equiv 0 \pmod{2,5}, 1 \pmod{3,7}$.

Solve $k \equiv 0 \pmod{10}$ and $1 \pmod{3}, 1 \pmod{7}$.

$k \equiv 0 \pmod{10}$ and $1 \pmod{3}$: $k=10m \equiv 1 \pmod{3} \Rightarrow 10m \equiv 1 \pmod{3} \Rightarrow m \equiv 1 \pmod{3}$. $m=3n+1$. $k=10*(3n+1)=30n+10 \equiv 10 \pmod{30}$.

Then solve $k \equiv 10 \pmod{30}$ and $1 \pmod{7}$.

$k=30n+10 \equiv 1 \pmod{7} \Rightarrow 30n+10 \equiv 1 \pmod{7} \Rightarrow 30 \equiv 2 \pmod{7}$, so $2n+10 \equiv 1 \pmod{7} \Rightarrow 2n \equiv -9 \equiv -2 \equiv 5 \pmod{7} \Rightarrow n \equiv (5)(\text{inverse of } 2 \pmod{7})=54=20 \equiv 6 \pmod{7}$. $n=7p+6$. Then $k=30*(7p+6)+10=210p+190$. Thus, $k \equiv 190 \pmod{210}$.

Check $k=190$:

$(190^2 - 190)/210 = (36100 - 190)/210 = 35910/210 = 171$. $d_1=171$, $d_2=190 - 171=19$. $n=36100$. In base 211, $36100 = 171*211 + 19 = 36081 + 19 = 36100$. Digits 171 and 19 sum to 190. Correct.

Thus, $k=190$ is a solution. Similarly, $k=190$ is in the interval.

Similarly, another combination:

$k \equiv 0 \pmod{2,7,1} \pmod{3,5}$.

Solve $k \equiv 0 \pmod{14}$ and $1 \pmod{3,1} \pmod{5}$.

$k \equiv 0 \pmod{14}$ and $1 \pmod{3}$: $14m \equiv 1 \pmod{3} \Rightarrow 14 \equiv 2 \pmod{3}$, so $2m \equiv 1 \pmod{3} \Rightarrow m \equiv 2 \pmod{3}$. $m=3n+2$. $k=14*(3n+2)=42n+28 \equiv 28 \pmod{42}$.

Then solve $k \equiv 28 \pmod{42}$ and $1 \pmod{5}$.

$k=42n+28 \equiv 1 \pmod{5} \Rightarrow 42n+28 \equiv 1 \pmod{5} \Rightarrow 42 \equiv 2 \pmod{5}$, $28 \equiv 3 \pmod{5}$. So $2n+3 \equiv 1 \pmod{5} \Rightarrow 2n \equiv -2 \equiv 3 \pmod{5} \Rightarrow n \equiv (3)(\text{inverse of } 2)=33=9 \equiv 4 \pmod{5}$. $n=5p+4$. Then $k=42*(5p+4)+28=210p+176+28=210p+204$. Wait, $42*4=168$, $168+28=196$. Wait, maybe miscalculation.

等等, $k = 42n + 28 \equiv 1 \pmod{5}$.

Compute $42n+28 \pmod{5}$:

$42 \equiv 2 \pmod{5}$, $28 \equiv 3 \pmod{5}$. So equation: $2n+3 \equiv 1 \pmod{5} \Rightarrow 2n \equiv -2 \equiv 3 \pmod{5} \Rightarrow n \equiv (3)(\text{inverse of } 2)=33=9 \equiv 4 \pmod{5}$.

Therefore, $n=5p+4$. Then $k=42*(5p+4)+28=210p+168+28=210p+196 \equiv 196 \pmod{210}$.

Check $k=196$:

$(196^2 - 196)/210 = (38416 - 196)/210 = 38220/210 = 182$. $d_1=182$, $d_2=196 - 182=14$. $n=38416$. In base 211, $38416 = 182*211 + 14 = 38302 + 14 = 38416$. Digits 182 and 14 sum to 196. Correct.

Thus, $k=196$ is a solution.

Therefore, the solutions in $[15,210]$ are:

30,36,70,84,90,105,120,126,150,156,180,186,190,196. That's 14 solutions. Therefore, 14 solutions. Each corresponding to a combination of choosing 0 or 1 mod each prime. Therefore, each solution is unique, and they are all ≥ 15 . Therefore, in base 211, there are 14 beautiful numbers, which is more than ten. Therefore, the minimal b is 211.

But let's check if there is a smaller b with more solutions. For example, take $(b-1)=235 \cdot 11=330$, so $b=331$. Then the number of solutions would be $2^4=16$. However, $\text{ceil}(\sqrt{331}) \approx 18.3$, so interval $[19, 330]$. The solutions mod 330 would be 16 numbers. If all are ≥ 19 , then 16 solutions. But likely, some solutions are less than 19, so the count would be 16 minus those less than 19. But this might result in more than 14 solutions. However, this would require checking.

Alternatively, take $(b-1)=2357=210$, $b=211$, which gives 14 solutions. The next candidate would be $(b-1)=2357 \cdot 11=2310$, $b=2311$, giving 32 solutions. But this is much larger.

Alternatively, check $b=105$. Then $(b-1)=104=813$. The number of solutions would be related to the factors of 8 and 13. Since $8=2^3$ and 13 is prime. The number of solutions mod 8 is $\phi(8)=4$? Wait, no. For modulus 8, the number of solutions to $k(k-1) \equiv 0 \pmod{8}$ is 2 (as $k \equiv 0$ or $1 \pmod{8}$). Similarly, for modulus 13, the number of solutions is 2. Therefore, total solutions mod $104=813$ is $2 \cdot 2=4$. Therefore, in the interval $[\text{ceil}(\sqrt{105}) \approx 11, 104]$, the number of solutions is 4. Thus, four solutions. Which is less than ten.

Therefore, the minimal b is 211.

Final Answer

$\boxed{211}$

<|end_of_thought|>

<|begin_of_solution|>

To determine the least integer $(b \geq 2)$ for which there are more than ten (b) -beautiful integers, we need to find a base (b) where there are more than ten two-digit numbers (n) in base (b) such that the sum of their digits equals $(\sqrt{n})$.

A number (n) is (b) -beautiful if:

1. (n) is a two-digit number in base (b) , i.e., $(b \leq n \leq b^2 - 1)$.
2. The sum of the digits of (n) in base (b) equals $(\sqrt{n})$.

Key steps:

1. For (n) to be a two-digit number in base (b) , (n) must be a perfect square (k^2) where (k) is an integer such that $(\lceil \sqrt{b} \rceil \leq k \leq b - 1)$.
2. The digits of (k^2) in base (b) must sum to (k) . This translates to the condition that $(k^2 - k)$ is divisible by $(b - 1)$.

To find the smallest (b) such that there are more than ten such (k) , we need $(b - 1)$ to have many divisors.

The number of solutions is related to the number of integers (k) such that $(b - 1)$ divides $(k(k - 1))$.

By analyzing the problem, we find that choosing $(b - 1)$ as the product of the first few primes (square-free) maximizes the number of solutions. The smallest such (b) is found when $(b - 1 = 2 \cdot 3 \cdot 5 \cdot 7 = 210)$, leading to $(b = 211)$.

In this case, the number of solutions (beautiful numbers) is determined by the combinations of residues modulo the prime factors of 210. This results in 16 solutions modulo 210, all of which are valid in the range $\lceil \sqrt{211} \rceil = 15$ to 210, giving us 14 valid solutions (more than ten).

Thus, the least integer $b \geq 2$ for which there are more than ten b -beautiful integers is $\boxed{211}$.

<|end_of_solution|>

GT: 211

Predict: 211 