

Question

(AIME24 Question No.24)

Find the number of ways to place a digit in each cell of a 2×3 grid so that the sum of the two numbers formed by reading left to right is 999, and the sum of the three numbers formed by reading top to bottom is 99. The grid below is an example of such an arrangement because $8 + 991 = 999$ and $9 + 9 + 81 = 99$.

0	0	8
9	9	1

Qwen-2.5-7B Answer

Response Length: 967

To solve this problem, we need to determine the number of ways to place digits in a 2×3 grid such that the sum of the two numbers formed by reading left to right is 999, and the sum of the three numbers formed by reading top to bottom is 99. Let's break this down step by step.

1. Define the grid:

Let the grid be represented as:

$$\begin{array}{ccc} a & b & c \\ d & e & f \end{array}$$

The numbers formed by reading left to right are $100a + 10b + c$ and $100d + 10e + f$. The numbers formed by reading top to bottom are $10a + d$, $10b + e$, and $10c + f$.

2. Set up the equations:

- The sum of the two numbers formed by reading left to right is 999:

$$100a + 10b + c + 100d + 10e + f = 999$$

- The sum of the three numbers formed by reading top to bottom is 99:

$$10a + d + 10b + e + 10c + f = 99$$

3. Simplify the equations:

- From the first equation:

$$100a + 10b + c + 100d + 10e + f = 999$$

This implies:

$$a + d = 9, b + e = 9, c + f = 9$$

- From the second equation:

$$10a + d + 10b + e + 10c + f = 99$$

This is consistent with the previous simplification since:

$$10(a + b + c) + (d + e + f) = 99$$

Given $a + d = 9$, $b + e = 9$, and $c + f = 9$, it follows that:

$$10(a + b + c) + (9 - a) + (9 - b) + (9 - c) = 99$$

Simplifying, we get:

$$10(a + b + c) + 27 - (a + b + c) = 99$$

$$9(a + b + c) + 27 = 99$$

$$9(a + b + c) = 72$$

$$a + b + c = 8$$

Since $a + d = 9$, $b + e = 9$, and $c + f = 9$, and $a + b + c = 8$, it follows that:

$$d = 9 - a, e = 9 - b, f = 9 - c$$

4. Determine the possible values:

- a, b, c must be digits (0-9) such that $a + b + c = 8$.

- For each valid combination of a, b, c , d, e, f are uniquely determined as $d = 9 - a$, $e = 9 - b$, $f = 9 - c$.

5. Count the combinations:

- We need to count the number of non-negative integer solutions to $a + b + c = 8$ with $0 \leq a, b, c \leq 9$.

- This is a classic "stars and bars" problem with an additional constraint. The number of solutions is given by the binomial coefficient:

$$\binom{8 + 3 - 1}{3 - 1} = \binom{10}{2} = 45$$

Therefore, the number of ways to place the digits in the grid is 45.

Figure 21: Qwen2.5-Base-7B Correct Response - Case 2.

D. Prompt Templates

We provide the prompt templates used for training and evaluation in our experiments. The prompt for SimpleRL training and evaluation is shown in Figure 22, while the prompt for Oat-Zero is shown in Figure 23. For Code-R1 training, prompt in Figure 24 is adopted. For Code-R1 evaluation, we follow the original codebase and adopt the default templates from the benchmarks, including LiveCodeBench prompt (Figure 25), HumanEval+, and MBPP+ prompt (Figure 26). The prompt used for EasyR1 training and evaluation is shown in Figure 27. For VerL-trained RL models, as discussed in Section 4.3 and Section 4.4, the training and evaluation prompts are provided in Figure 28. For evaluating Mistral and Magistral models on AIME24/25, prompts are provided in Figure 29. To ensure a fair comparison, the base models use the same prompts as their corresponding RL-trained counterparts during evaluation.

SimpleRL Prompt

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{question}
Please reason step by step, and put your final answer within\\boxed{{ }}.<|im_end|>
<|im_start|>assistant
```

Figure 22: Prompt for SimpleRL Training and Evaluation. The base model uses the same prompt as the RL model during evaluation.

Oat Prompt

```
<|im_start|>system
Please reason step by step, and put your final answer within \\boxed{{ }}.<|im_end|>
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
```

Figure 23: Prompt for Oat-Zero training and evaluation.

Code-R1 Prompt

```
<|im_start|>system
You are a helpful programming assistant. The user will ask you a question and you
as the assistant solve it. The assistant first thinks how to solve the task through
reasoning and then provides the user with the final answer. The reasoning process
and answer are enclosed within <think>...</think> and <answer>...</answer> tags,
respectively.<|im_end|>
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
```

Figure 24: Prompt for Code-R1 training.

LiveCodeBench (Code Generation) Prompt

```

<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
You will be given a question (problem specification) and will generate a correct
Python program that matches the specification and passes all tests. You will NOT
return anything except for the program.

Question: {question.question_content}

{ if question.starter_code }
### Format:
{PromptConstants.FORMAT_MESSAGE_WITH_STARTER_CODE}
```python
{question.starter_code}
```

<|im_end|>
{ else }
### Format:
{PromptConstants.FORMAT_MESSAGE_WITHOUT_STARTER_CODE}
```python
YOUR CODE HERE
```

<|im_end|>
<|im_start|>assistant
```python

```

Figure 25: Since Code-R1 does not specify an evaluation prompt, we adopt the original LiveCodeBench evaluation prompt. To encourage both the base and RL-trained models to generate code, we append “python” to the end of the prompt. Using this setup, we reproduce a pass@1 score of 28.6, which is close to the reported 29.7.

#### HumanEval+ & MBPP+ Prompt

```

<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
Please provide a self-contained Python script that solves the following problem in a
markdown code block:
```
{python_task_prompt}
```

<|im_end|>
<|im_start|>assistant
Below is a Python script with a self-contained function that solves the problem and
passes corresponding tests:
```python

```

Figure 26: Prompt for Code-R1 Evaluation on HumanEval+ and MBPP+.