

本次 lab 中选择的题目是“黑洞数”，即指于四位数中，只要数字不完全相同，将数字由大到小的排列减去由小到大的排列，通过前者减去后者的操作，在七步之内都会对应到 6174。这种现象类似黑洞（进去后就出不来了），故称为黑洞数。

【30 分】自定一个问题，编写代码，用上下文提示学习方法，通过大模型得到答案。

```
import openai

# 启用OpenAI库
client = openai.OpenAI(base_url="http://localhost:11434/v1",
                       api_key="ollama") # 提前安装 ollama

# 定义 24 点游戏 函数
def solve(problem):
    # 24 点游戏的规则
    SYSTEM_MESSAGE = f"""
    黑洞数是指于四位数中，只要数字不完全相同，将数字由大到小的排列减去由小到大的排列，通过前者减去后者的操作，在七步之内都会对应到6174。
    这种现象类似黑洞（进去后就出不来了），故称为黑洞数。
    请输入任意4个不同的数字:
    """

    # 24 点游戏的例子，以及对应的答案
    EXAMPLE1_INPUT = f"""请输入任意4个不同的数字: \n9, 8, 9, 1"""
    EXAMPLE1_OUTPUT = f"f(9891)=9981-1899=8082, f(8082)=8820-0288=8532, f(8532)=8532-2358=6174, f(6174)=7641-1467=6174"

    EXAMPLE2_INPUT = f"""请输入任意4个不同的数字: \n4, 3, 2, 1"""
    EXAMPLE2_OUTPUT = f"f(4321)=4321-1234=3087, f(3087)=8730-0378=8352, f(8352)=8532-2358=6174"

    EXAMPLE3_INPUT = f"""请输入任意4个不同的数字: \n1238"""
    EXAMPLE3_OUTPUT = f"f(1238)=8321-1238=7083, f(7083)=8730-378=8352, f(8352)=8532-2358=6174"

    # 24 点游戏的问题
    PROBLEM = f"""黑洞数是指于四位数中，只要数字不完全相同，将数字由大到小的排列减去由小到大的排列，
    通过前者减去后者的操作，在七步之内都会对应到6174。这种现象类似黑洞（进去后就出不来了），故称为黑洞数。
    请输入任意4个不同的数字: \n{problem}"""

    # 调用OpenAI的聊天接口
    chat_response = client.chat.completions.create(
        model="qwen2.5:3b", # 在cmd命令行里运行: ollama pull qwen2.5:3b
        messages=[
            # 传入例子和问题
            {"role": "system", "content": f"""{SYSTEM_MESSAGE}"""},
            {"role": "user", "content": f"""{EXAMPLE1_INPUT}"""},
            {"role": "assistant", "content": f"""{EXAMPLE1_OUTPUT}"""},
            {"role": "user", "content": f"""{EXAMPLE2_INPUT}"""},
            {"role": "assistant", "content": f"""{EXAMPLE2_OUTPUT}"""},
            {"role": "user", "content": f"""{EXAMPLE3_INPUT}"""},
            {"role": "assistant", "content": f"""{EXAMPLE3_OUTPUT}"""},
            {"role": "user", "content": f"""{PROBLEM}"""}, ],
        temperature=0.5, # 设置生成文本的创造性(0~1). 0.5为默认值, 越大越有创造性
        top_p=0.8, # 设置生成文本的多样性(0~1). 0.8为默认值, 越大越多样
        max_tokens=512, # 设置生成文本的最大长度(1~4096)
        extra_body={"repetition_penalty": 1.05}, # 重复惩罚, 1.05为默认值, 越大越不重复
    )

    # print("Chat response:", chat_response)
    return chat_response.choices[0].message.content

# 测试
print(solve("0, 2, 3, 4"))
```

结果：

好的，我们来计算一下这个四位数的黑洞过程。

给定的四位数是：0234

首先将数字由大到小排列：4320

然后将数字由小到大排列：0234

接下来进行减法操作：

$$\backslash[4320 - 0234 = 4086 \backslash]$$

现在我们用4086继续这个过程：

首先将数字由大到小排列：8640

然后将数字由小到大排列：0468

接下来进行减法操作：

$$\backslash[8640 - 0468 = 8172 \backslash]$$

继续这个过程：

首先将数字由大到小排列：8721

然后将数字由小到大排列：1278

接下来进行减法操作：

$$\backslash[8721 - 1278 = 7443 \backslash]$$

...

接下来进行减法操作：

$$\backslash[7641 - 1467 = 6174 \backslash]$$

经过七步，我们已经得到了黑洞数6174。

【30 分】自定一个问题，编写代码，用思维链提示方法，通过大模型得到答案。

```
import openai

client = openai.OpenAI(base_url="http://localhost:11434/v1", api_key="ollama")

def solve():
    SYSTEM_MESSAGE = f"""
    你是一个数学问题的专家，你的目标是帮助用户解决以下问题。
    请提供一个解决问题的详细步骤，一步一步解释你的思路，最后给出一个解答。
    """

    PROBLEM = f"黑洞数是指于四位数中，只要数字不完全相同，将数字由大到小的排列减去由小到大的排列，通过前者减去后者的操作，在七步之内都会对应到6174。"

    chat_response = client.chat.completions.create(
        model="qwen2.5:3b",
        messages=[
            {"role": "system", "content": f"{SYSTEM_MESSAGE}"},
            {"role": "user", "content": f"{PROBLEM}"},
        ],
        temperature=0.5,
        top_p=0.8,
        max_tokens=2048,
        extra_body={"repetition_penalty": 1.05},
    )
    # print("Chat response:", chat_response)
    return chat_response.choices[0].message.content

print(solve())
```

结果：

当输入一个四位数时，该四位数由大到小排列和由小到大的排列之差即为这个数减去它反转后的数。  
如：1357-7531=-6174。  
所以将1357变为黑洞数的过程如下：  
第一步：1357-7531=-6174  
第二步：6174-4716=-2580  
第三步：8520-0258=8262  
第四步：8622-2268=6354  
第五步：6543-3456=3087  
第六步：8730-0378=8352  
第七步：8532-2358=6174  
即输入的四位数1357经过七步就会变为黑洞数6174。  
若输入其他数字，例如：1234，则为：  
第一步：4321-1234=3087  
第二步：8730-0378=8352  
第三步：8532-2358=6174

【40 分】自定一个问题，用 LongChain 编写代码，通过大模型得到答案。

```
from langchain_community.llms import Ollama # 本地模型
from langchain_core.prompts import ChatPromptTemplate, FewShotChatMessagePromptTemplate
from langchain_core.output_parsers import StrOutputParser

# 1. 定义模型
llm = Ollama(model="qwen2.5:3b", base_url="http://localhost:11434") # 确保已运行 ollama serve

# 2. 定义提示词模板 (含示例)
examples = [
    {
        "problem": "黑洞数是指于四位数中,只要数字不完全相同,将数字由大到小的排列减去由小到大的排列,通过前者减去后者的操作,在七步之内都会",
        "solution": "f(9891)=9981-1899=8082,f(8082)=8820-0288=8532,f(8532)=8532-2358=6174,f(6174)=7641-1467=6174",
    },
]

# 定义示例提示词模板
example_prompt = ChatPromptTemplate.from_messages([
    ("human", "{problem}"),
    ("ai", "{solution}")
])

# 用来生成示例
few_shot_prompt = FewShotChatMessagePromptTemplate(
    example_prompt=example_prompt,
    examples=examples
)

# 定义完整的提示词模板
full_prompt = ChatPromptTemplate.from_messages([
    ("system", "你是一个数学问题的专家,请一步步思考并给出详细解答。"),
    few_shot_prompt,
    ("human", "{problem}")
])

# 3. 定义链 (Chain): 提示词 + 模型 + 输出解析
chain = full_prompt | llm | StrOutputParser() # 默认使用 JSON 输出解析器

# 4. 调用链解决问题
result = chain.invoke({"problem": "黑洞数是指于四位数中,只要数字不完全相同,将数字由大到小的排列减去由小到大的排列,通过前者减去后者的操作"})
print(result)
```

结果:

为了找到数字 2035 在经过黑洞数过程所需的步骤,我们需要遵循以下步骤:每次将这个四位数的各个数字按从大到小排列,

我们用给定的例子为参考:

- 对于2035:
  - 大到小排列是:5320
  - 小到大排列是:0235 (记得首位不能是零)
  - 所以第一次操作的结果是:5320 - 235 = 4985

接下来我们重复这个过程:

- 大到小排列是:8954
- 小到大排列是:4589
- 第二次操作结果是:8954 - 4589 = 4365

继续进行:

- 大到小排列是:6543
- 小到大排列是:3456
- 第三次操作结果是:6543 - 3456 = 3087

接下来的操作:

- 大到小排列是：8730
- 小到大排列是：0378（首位不能为零）
- 第四次操作结果是： $8730 - 378 = 8352$

继续进行：

- 大到小排列是：8532
- 小到大排列是：2358
- 第五次操作结果是： $8532 - 2358 = 6174$

现在我们已经找到了黑洞数 6174，根据题目要求，在七步之内完成。

所以对于输入数字 2035 过程如下：

1.  $2035 \rightarrow 5320 - 235 = 4985$
2.  $8954 - 4589 = 4365$
3.  $6543 - 3456 = 3087$
4.  $8730 - 378 = 8352$
5.  $8532 - 2358 = 6174$

所以输入数字 2035，经过五步后可以找到黑洞数 6174。