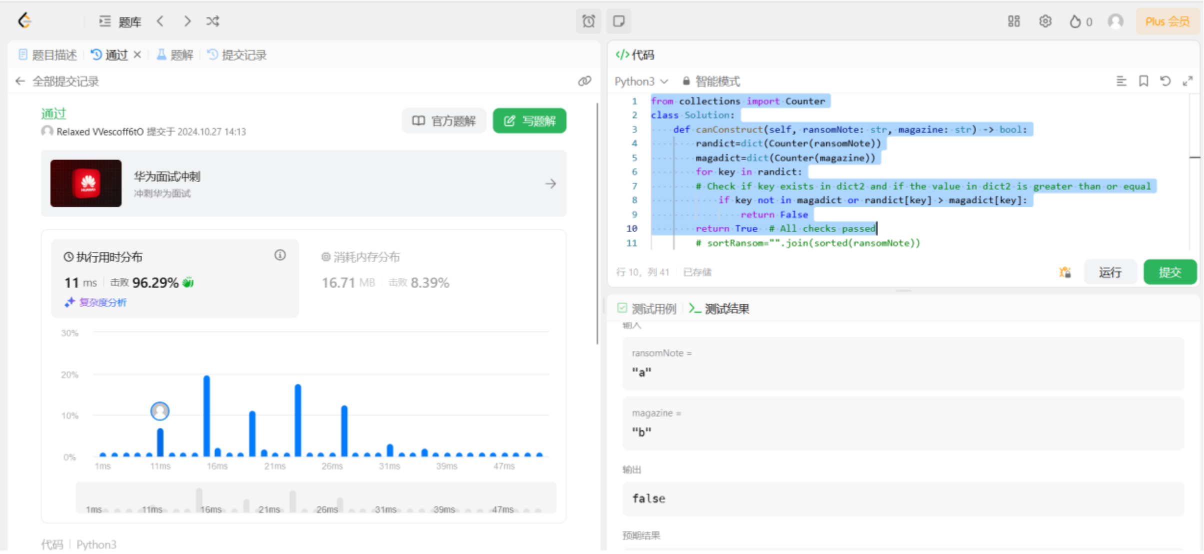
Task 1 LeetCode 383:

```
from collections import Counter

class Solution:

    def canConstruct(self, ransomNote: str, magazine: str) -> bool:

        randict=dict(Counter(ransomNote))

        magadict=dict(Counter(magazine))

        for key in randict:

        # Check if key exists in dict2 and if the value in dict2 is greater than or equal

            if key not in magadict or randict[key] > magadict[key]:

                return False

        return True   # All checks passed
```



通过

Relaxed Wescoff6tO 提交于 2024.10.27 14:13

执行用时分布

11 ms | 击败 96.29%

消耗内存分布

16.71 MB | 击败 8.39%

```
from collections import Counter
class Solution:
    def canConstruct(self, ransomNote: str, magazine: str) -> bool:
        randict=dict(Counter(ransomNote))
        magadict=dict(Counter(magazine))
        for key in randict:
        # Check if key exists in dict2 and if the value in dict2 is greater than or equal
            if key not in magadict or randict[key] > magadict[key]:
                return False
        return True  # All checks passed
        # sortRansom="".join(sorted(ransomNote))
```

行 10，列 41 | 已存储

输入

ransomNote =

```
"a"
```

magazine =

```
"b"
```

输出

```
false
```

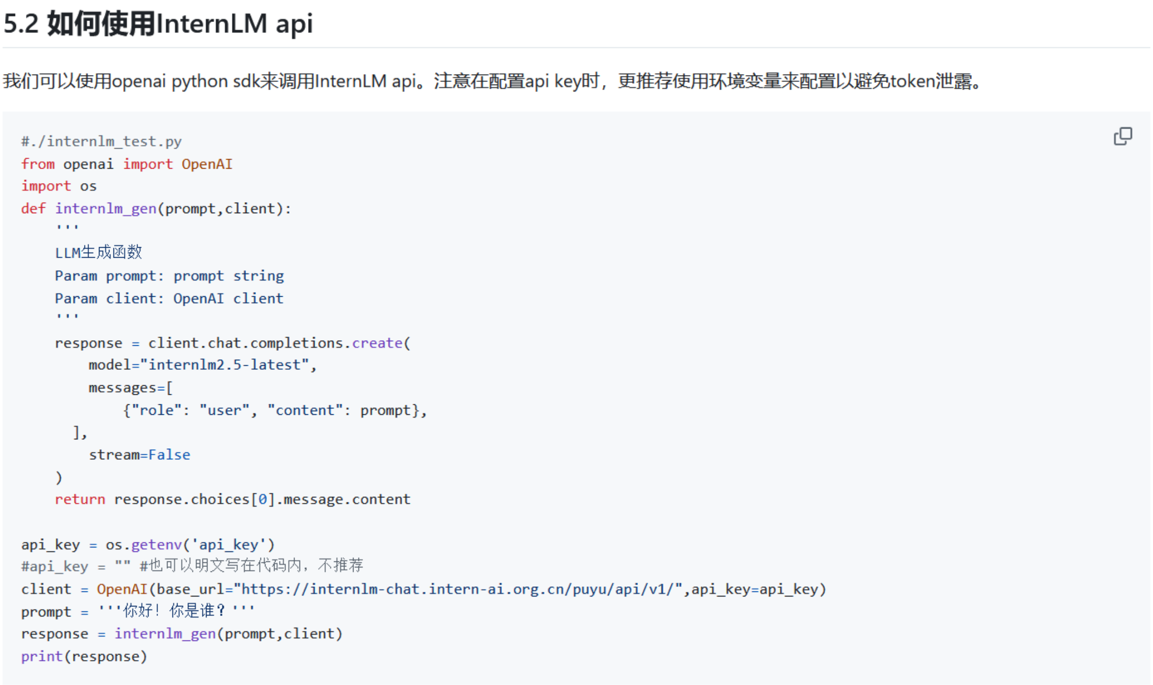
预期结果

Idea for code:

Create a dictionary for ransomNote and magazine. We first need to make sure all keys in the ransomNote dict appears in magazine, then if this condition passes, we need to make sure the number of times which each key appears in ransomNote must be less or equal to the number of appearances in the magazine dict. If both conditions are true, return true.

Task 2

第二个 task 没有用到 debug tool，我比较了一下 tutorial 里面用到的 code (Fig.1)发现 res_json = json.loads(res) 是多余的，加上报错信息也在第 30 行 (Fig.2). comment 之后无报错，结果如图三所示，InternLM 成功回答了 prompt 的内容

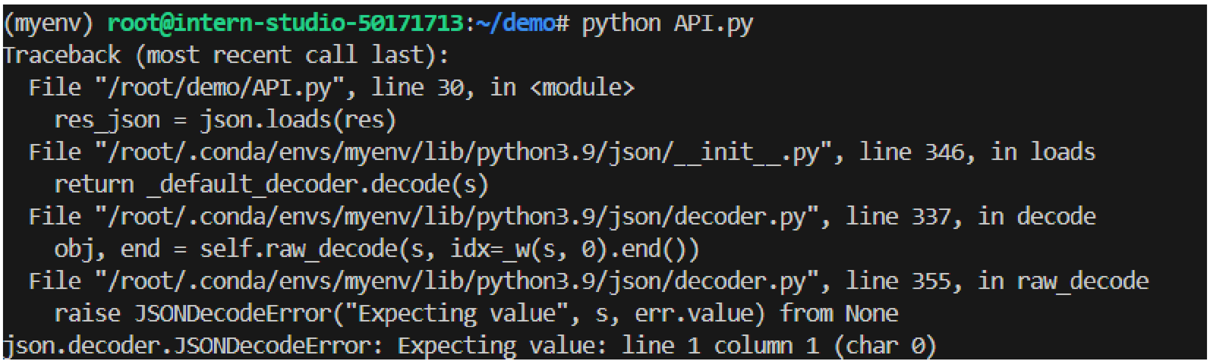


5.2 如何使用InternLM api

我们可以使用openai python sdk来调用InternLM api。注意在配置api key时，更推荐使用环境变量来配置以避免token泄露。

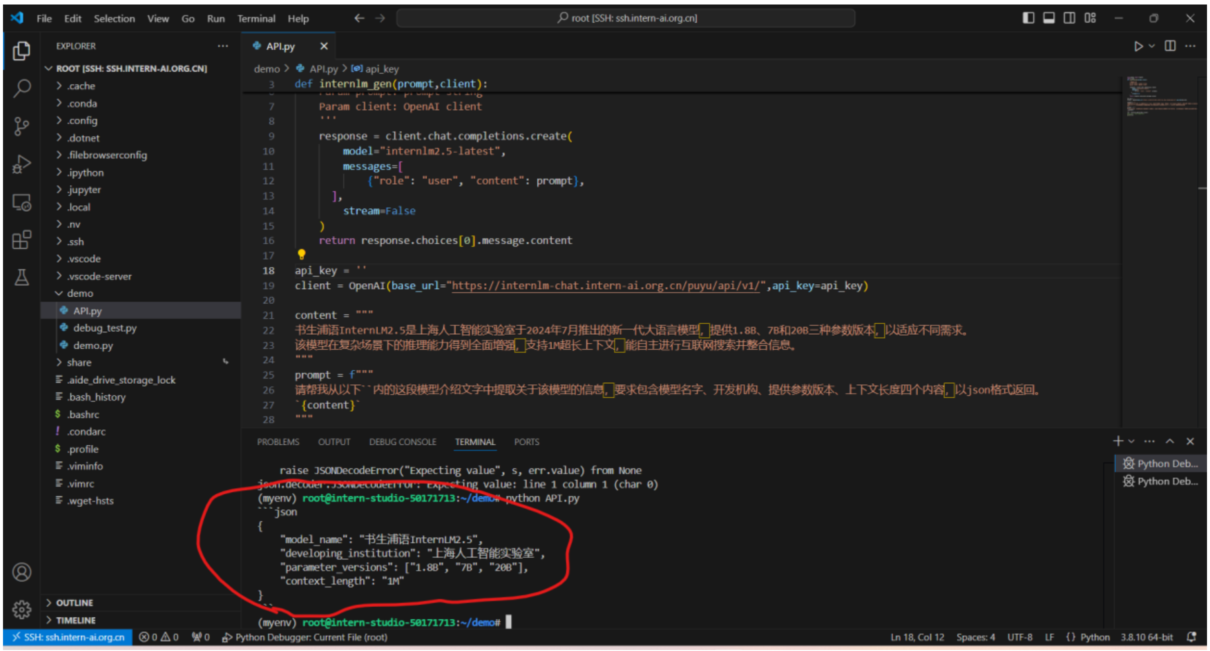
```
#./internlm_test.py
from openai import OpenAI
import os
def internlm_gen(prompt,client):
    '''
    LLM生成函数
    Param prompt: prompt string
    Param client: OpenAI client
    '''
    response = client.chat.completions.create(
        model="internlm2.5-latest",
        messages=[
            {"role": "user", "content": prompt},
      ],
        stream=False
    )
    return response.choices[0].message.content

api_key = os.getenv('api_key')
#api_key = "" #也可以明文写在代码内，不推荐
client = OpenAI(base_url="https://internlm-chat.intern-ai.org.cn/puyu/api/v1/",api_key=api_key)
prompt = '''你好！你是谁？'''
response = internlm_gen(prompt,client)
print(response)
```

Fig 1.



```
(myenv) root@intern-studio-50171713:~/demo# python API.py
Traceback (most recent call last):
  File "/root/demo/API.py", line 30, in <module>
    res_json = json.loads(res)
  File "/root/.conda/envs/myenv/lib/python3.9/json/__init__.py", line 346, in loads
    return _default_decoder.decode(s)
  File "/root/.conda/envs/myenv/lib/python3.9/json/decoder.py", line 337, in decode
    obj, end = self.raw_decode(s, idx=_w(s, 0).end())
  File "/root/.conda/envs/myenv/lib/python3.9/json/decoder.py", line 355, in raw_decode
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
```

Fig 2



```python
def internlm_gen(prompt,client):
    Param client: OpenAI client
    '''
    response = client.chat.completions.create(
        model="internlm2.5-latest",
        messages=[
            {"role": "user", "content": prompt},
        ],
        stream=False
    )
    return response.choices[0].message.content

api_key = ''
client = OpenAI(base_url="https://internlm-chat.intern-ai.org.cn/puyu/api/v1/",api_key=api_key)

content = """
书生浦语InternLM2.5是上海人工智能实验室于2024年7月推出的新一代大语言模型，提供1.8B、7B和20B三种参数版本，以适应不同需求。
该模型在复杂场景下的推理能力得到全面增强，支持1M超长上下文，能自主进行互联网搜索并整合信息。
"""

prompt = f"""
请帮我从以下``内的这段模型介绍文字中提取关于该模型的信息，要求包含模型名字、开发机构、提供参数版本、上下文长度四个内容，以json格式返回。
`{content}`
"""
```

~~~
    raise JSONDecodeError("Expecting value", s, err.value) from None
json.decoder.JSONDecodeError: Expecting value: line 1 column 1 (char 0)
(myenv) root@intern-studio-50171713:~/demo# python API.py
```json
{
    "model_name": "书生浦语InternLM2.5",
    "developing_institution": "上海人工智能实验室",
    "parameter_versions": ["1.8B", "7B", "20B"],
    "context_length": "1M"
}
```
(myenv) root@intern-studio-50171713:~/demo# 
~~~

Fig 3