1. 题目

1078: Bigram分词

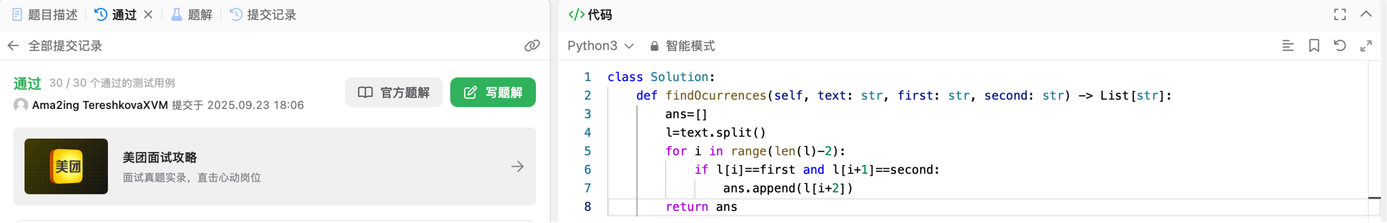
https://leetcode.cn/problems/occurrences-after-bigram/

思路：implementation

代码：

```
class Solution:
    def findOcurrences(self, text: str, first: str, second: str) -> List[str]:
        ans=[]
        l=text.split()
        for i in range(len(l)-2):
            if l[i]==first and l[i+1]==second:
                ans.append(l[i+2])
        return ans
```

代码运行截图 （至少包含有"Accepted"）



通过 30 / 30 个通过的测试用例

Ama2ing TereshkovaXVM 提交于 2025.09.23 18:06

```
class Solution:
    def findOcurrences(self, text: str, first: str, second: str) -> List[str]:
        ans=[]
        l=text.split()
        for i in range(len(l)-2):
            if l[i]==first and l[i+1]==second:
                ans.append(l[i+2])
        return ans
```

283.移动零

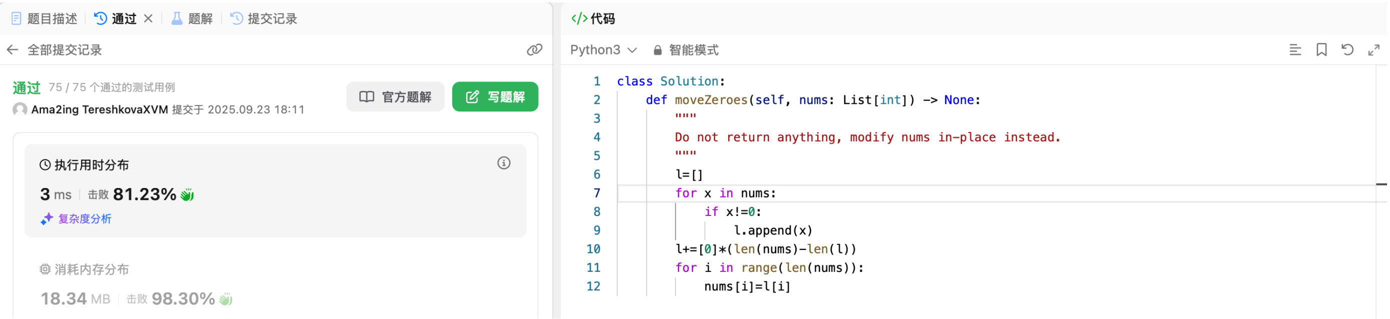
stack, two pinters, https://leetcode.cn/problems/move-zeroes/

思路：implementation

代码：

```
class Solution:
    def moveZeroes(self, nums: List[int]) -> None:
        """
        Do not return anything, modify nums in-place instead.
        """
        l=[]
        for x in nums:
            if x!=0:
                l.append(x)
        l+=[0]*(len(nums)-len(l))
        for i in range(len(nums)):
            nums[i]=l[i]
```

代码运行截图 （至少包含有"Accepted"）



通过 75 / 75 个通过的测试用例

Ama2ing TereshkovaXVM 提交于 2025.09.23 18:11

执行用时分布

3 ms | 击败 81.23%

消耗内存分布

18.34 MB | 击败 98.30%

```
class Solution:
    def moveZeroes(self, nums: List[int]) -> None:
        """
        Do not return anything, modify nums in-place instead.
        """
        l=[]
        for x in nums:
            if x!=0:
                l.append(x)
        l+=[0]*(len(nums)-len(l))
        for i in range(len(nums)):
            nums[i]=l[i]
```

20.有效的括号

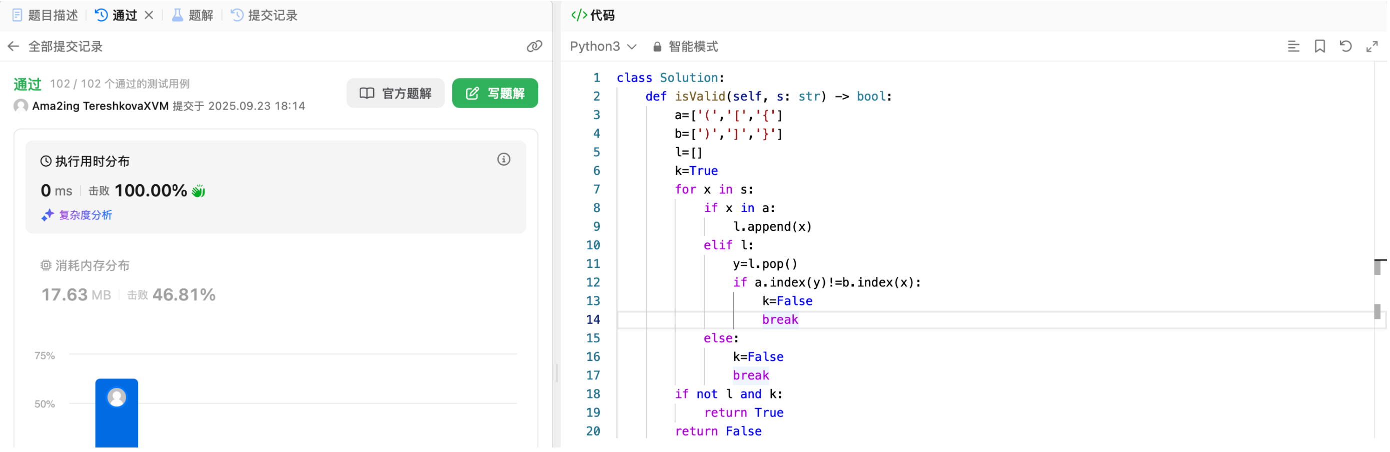
stack, https://leetcode.cn/problems/valid-parentheses/

思路：stack

代码：

```
class Solution:
    def isValid(self, s: str) -> bool:
        a=['(','[','{']
        b=[')',']','}']
        l=[]
        k=True
        for x in s:
            if x in a:
                l.append(x)
            elif l:
                y=l.pop()
                if a.index(y)!=b.index(x):
                    k=False
                    break
            else:
                k=False
                break
        if not l and k:
            return True
        return False
```

代码运行截图 （至少包含有"Accepted"）



通过 102 / 102 个通过的测试用例

Ama2ing TereshkovaXVM 提交于 2025.09.23 18:14

执行用时分布

0 ms | 击败 100.00%

复杂度分析

消耗内存分布

17.63 MB | 击败 46.81%

```
class Solution:
    def isValid(self, s: str) -> bool:
        a=['(','[','{']
        b=[')',']','}']
        l=[]
        k=True
        for x in s:
            if x in a:
                l.append(x)
            elif l:
                y=l.pop()
                if a.index(y)!=b.index(x):
                    k=False
                    break
            else:
                k=False
                break
        if not l and k:
            return True
        return False
```

118.杨辉三角

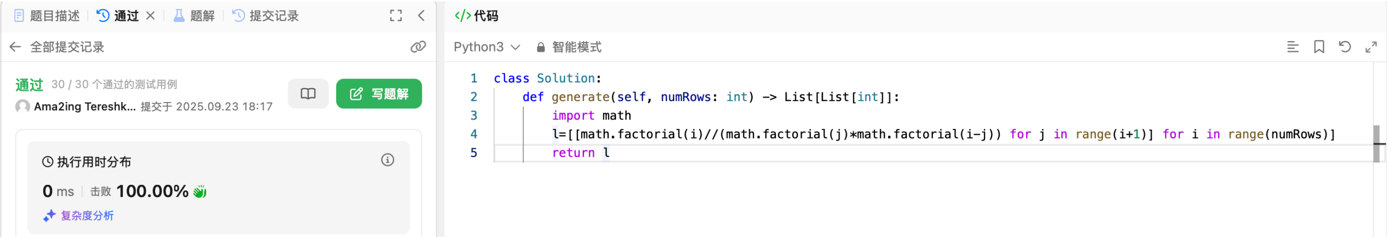
dp, https://leetcode.cn/problems/pascals-triangle/

思路：直接写公式

代码：

```
class Solution:
    def generate(self, numRows: int) -> List[List[int]]:
        import math
        l=[[math.factorial(i)//(math.factorial(j)*math.factorial(i-j)) for j in range(i+1)] for i in range(numRows)]
        return l
```

代码运行截图 （至少包含有"Accepted"）



通过 30 / 30 个通过的测试用例

Ama2ing Tereshk... 提交于 2025.09.23 18:17

执行用时分布

0 ms | 击败 100.00%

复杂度分析

```
class Solution:
    def generate(self, numRows: int) -> List[List[int]]:
        import math
        l=[[math.factorial(i)//(math.factorial(j)*math.factorial(i-j)) for j in range(i+1)] for i in range(numRows)]
        return l
```

46.全排列

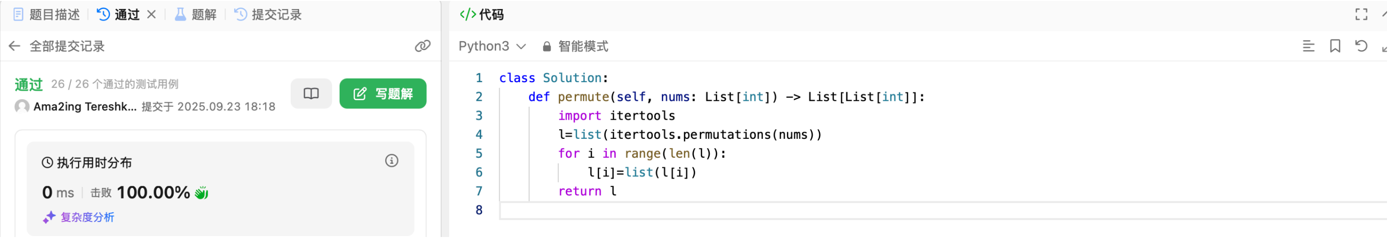
backtracking, https://leetcode.cn/problems/permutations/

思路：调包

代码

```python
class Solution:
    def permute(self, nums: List[int]) -> List[List[int]]:
        import itertools
        l=list(itertools.permutations(nums))
        for i in range(len(l)):
            l[i]=list(l[i])
        return l
```

（至少包含有"Accepted"）



通过 26 / 26 个通过的测试用例

Ama2ing Tereshk... 提交于 2025.09.23 18:18

执行用时分布

0 ms | 击败 100.00%

复杂度分析

```python
class Solution:
    def permute(self, nums: List[int]) -> List[List[int]]:
        import itertools
        l=list(itertools.permutations(nums))
        for i in range(len(l)):
            l[i]=list(l[i])
        return l
```

78.子集

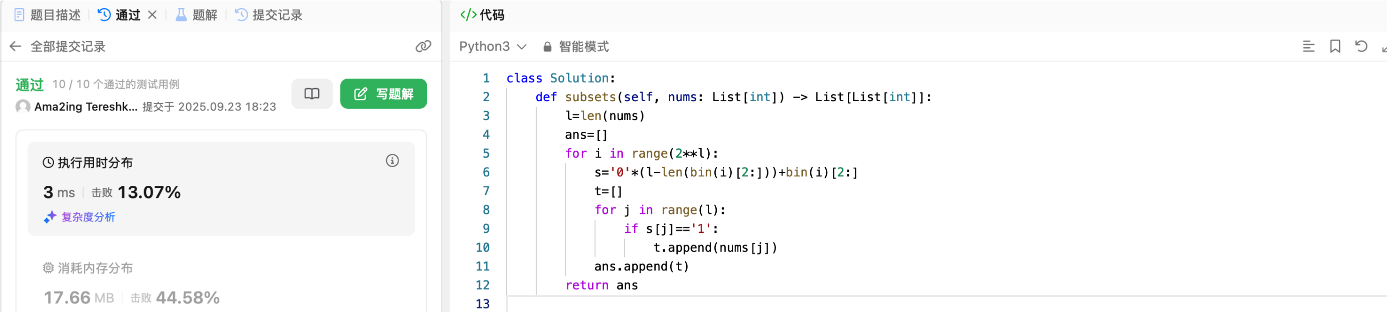
backtracking, https://leetcode.cn/problems/subsets/

思路：二进制

代码

```
class Solution:
    def subsets(self, nums: List[int]) -> List[List[int]]:
        l=len(nums)
        ans=[]
        for i in range(2**l):
            s='0'*(l-len(bin(i)[2:]))+bin(i)[2:]
            t=[]
            for j in range(l):
                if s[j]=='1':
                    t.append(nums[j])
            ans.append(t)
        return ans
```

（至少包含有"Accepted"）



通过 10 / 10 个通过的测试用例

Ama2ing Tereshk... 提交于 2025.09.23 18:23

执行用时分布

3 ms | 击败 13.07%

复杂度分析

消耗内存分布

17.66 MB | 击败 44.58%

```
class Solution:
    def subsets(self, nums: List[int]) -> List[List[int]]:
        l=len(nums)
        ans=[]
        for i in range(2**l):
            s='0'*(l-len(bin(i)[2:]))+bin(i)[2:]
            t=[]
            for j in range(l):
                if s[j]=='1':
                    t.append(nums[j])
            ans.append(t)
        return ans
```

2. 学习总结和个人收获

如果发现作业题目相对简单，有否寻找额外的练习题目，如“数算2025fall每日选做”、LeetCode、Codeforces、洛谷等网站上的题目。

忙忙忙忙忙。。。只能跟着每日选做。有些矛盾的想法，主要是觉得每日选做太简单，怕后面上强度跟不上，但提前上了强度也怕忙不过来。