

1. 题目

1078: Bigram分词

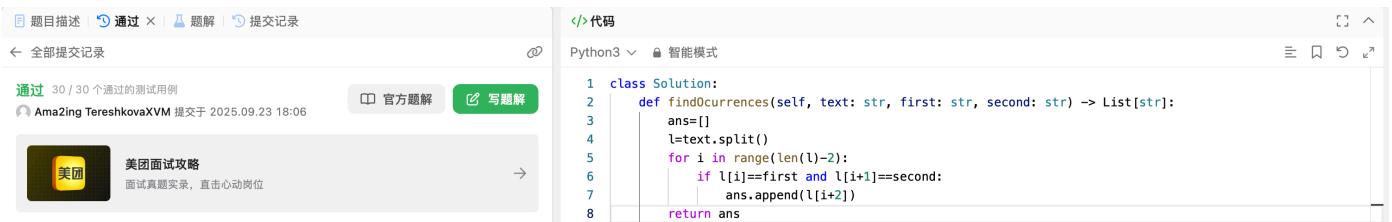
<https://leetcode.cn/problems/occurrences-after-bigram/>

思路: implementation

代码:

```
class Solution:
    def findOccurrences(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")



283. 移动零

stack, two pointers, <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")

题目描述 | 通过 × | 题解 | 提交记录

← 全部提交记录

通过 75 / 75 个通过的测试用例

Official Solution | Write Solution

Amaing TereshkovaXVM 提交于 2025.09.23 18:11

① 执行用时分布

3 ms | 击败 81.23% 🎉

复杂度分析

② 消耗内存分布

18.34 MB | 击败 98.30% 🎉

代码

Python3

智能模式

```
1 class Solution:
2     def moveZeroes(self, nums: List[int]) -> None:
3         """
4             Do not return anything, modify nums in-place instead.
5         """
6         l = []
7         for x in nums:
8             if x != 0:
9                 l.append(x)
10        l += [0] * (len(nums) - len(l))
11        for i in range(len(nums)):
12            nums[i] = l[i]
```

20.有效的括号

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

思路：stack

代码：

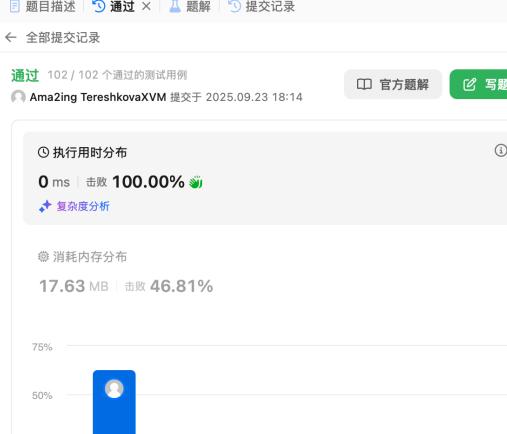
```
class Solution:
    def isValid(self, s: str) -> bool:
        a=['(', '[', '{']
        b=[')', ']', '}']
        l=[]
        k=True
        for x in s:
            if x in a:
                l.append(x)
            elif l:
                y=l.pop()
                if a.index(y)!=b.index(x):
                    k=False
                break
        return k
```

```

        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")



题目描述 | 通过 × | 题解 | 提交记录

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%

消耗内存分布
17.63 MB 击败 46.81%

代码

```

1 class Solution:
2     def isValid(self, s: str) -> bool:
3         a=['(',')','{']
4         b=[')',')','}']
5         l=[]
6         k=True
7         for x in s:
8             if x in a:
9                 l.append(x)
10            elif l:
11                y=l.pop()
12                if a.index(y)!=b.index(x):
13                    k=False
14                    break
15            else:
16                k=False
17                break
18        if not l and k:
19            return True
20        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")



题目描述 | 通过 × | 题解 | 提交记录

通过 30 / 30 个通过的测试用例

Ama2ing TereshkovaXVM 提交于 2025.09.23 18:17

执行用时分布
0 ms 击败 100.00%

代码

```

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

```

46.全排列

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

思路：调包

代码

```
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")

The screenshot shows a LeetCode submission page for problem 46. The code is pasted into the editor, and the output shows it has passed all 26 test cases. The execution time is 0 ms, and the success rate is 100.00%. The complexity analysis section is collapsed.

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")

The screenshot shows a LeetCode submission page. The top bar includes links for problem description, status, solution, and submission history. The main area displays the code in Python3 and its execution statistics. The code is identical to the one above. The statistics show 10/10 tests passed, a runtime of 3 ms (beating 13.07%), and a memory usage of 17.66 MB (beating 44.58%).

```

1 class Solution:
2     def subsets(self, nums: List[int]) -> List[List[int]]:
3         l=len(nums)
4         ans=[ ]
5         for i in range(2**l):
6             s='0'*(l-len(bin(i)[2:]))+bin(i)[2:]
7             t=[ ]
8             for j in range(l):
9                 if s[j]=='1':
10                     t.append(nums[j])
11             ans.append(t)
12         return ans
13

```

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。