

WEEK3

段雷

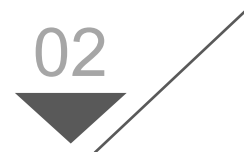
目录

CONTENTS



01

递归



02

本周趣题



03

题目讲解

01递归

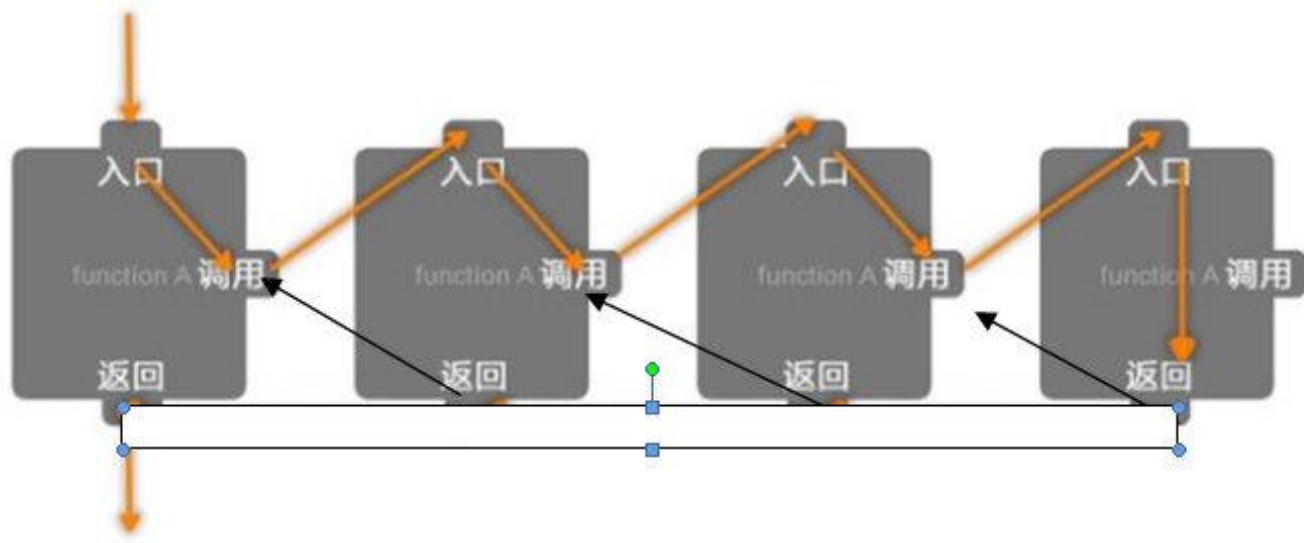
递归

从前有座山，山里有座庙，庙里有个和尚，和尚在讲故事，从前有座山，山里有座庙，庙里有个和尚，和尚在讲故事，从前有座山...

要理解递归，就得先了解什么是递归

吓得我抱起了

抱着抱着抱着我的小鲤鱼的我的我的我



递归

全排列问题:

给正整数n, 要求输出n的全排列

Sample Input:

3

Sample Output:

[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

八皇后问题:

给一个8*8的棋盘, 放置8个皇后, 要求输出所有皇后不会互相伤害的方案

全排列代码

```
n = 3
chosen = [False] * n
a = [0] * n
ans = []
def dfs(x):
    if x == n:
        ans.append(a.copy())
    for i in range(n):
        if chosen[i] == False:
            a[x] = i
            chosen[i] = True
            dfs(x + 1)
            chosen[i] = False
            a[x] = 0
dfs(0)
print(ans)
```

八皇后代码

```
n = 8
ans = []
a = [0] * n
chosen = [False] * n

def dfs(x):
    if x == n:
        ans.append(a.copy())
        return
    # a[x] = ?
    for i in range(n):
        if chosen[i]:
            continue
        # j : 0 ~ x - 1
        hurt_each_other = False
        for j in range(x):
            if j + a[j] == x + i:
                hurt_each_other = True
                break
            if j - a[j] == x - i:
                hurt_each_other = True
                break
        if hurt_each_other == False:
            a[x] = i
            chosen[i] = True
            dfs(x + 1)
            chosen[i] = False
            a[x] = 0
```

```
dfs(0)
print(ans)
```

八皇后代码（优化后）

```
n = 8
ans = []
a = [0] * n

def dfs(x):
    if x == n:
        ans.append(a.copy())
        return
    for i in range(n):
        hurt_each_other = False
        for j in range(x):
            if a[j] == i or j + a[j] == x + i or j - a[j] == x - i:
                hurt_each_other = True
                break
        if hurt_each_other == False:
            a[x] = i
            dfs(x + 1)
            a[x] = 0
```

```
dfs(0)
print(ans)
```

02本周趣题

GS OA

Every year, an investment firm invests into several companies shares Majority of their investments are 3 companies: A, B and C. As they invest, they create a string of all of their investments in order. Given a string of their investments, determine the number of time periods they invested in the three major companies shares.

For example, given the investments as $s = \text{"ABBCZBAC"}$, for total number of investments $n=7$. Starting from the left, the first substring that contains an investment in all companies is "ABBC" . There are 13 substrings of s which meet the criterion: $[\text{"ABBC"}, \text{"ABBCZ"}, \text{"ABBCZB"}, \text{"ABBCZBA"}, \text{"ABBCZBAC"}, \text{"BBCZBA"}, \text{"BBCZBAC"}, \text{"BCZBA"}, \text{"BCZBAC"}, \text{"CZBA"}, \text{"CZBAC"}, \text{"ZBAC"}, \text{"BAC"}]$

Note: Two substrings are considered different if the starting, ending, or both positions differ.

$1 \leq \text{length}(s) \leq 10^5$ $s[i]$ is in $['A' \dots 'Z']$

Leetcode3 (GS OA PLUS)

3. Longest Substring Without Repeating Characters

Medium

👍 6393

💬 369

♡ Favorite

🔗 Share

Given a string, find the length of the **longest substring** without repeating characters.

Example 1:

Input: "abcabcbb"

Output: 3

Explanation: The answer is "abc", with the length of 3.

Example 2:

Input: "bbbbbb"

Output: 1

Explanation: The answer is "b", with the length of 1.

Example 3:

Input: "pwwkew"

Output: 3

Explanation: The answer is "wke", with the length of 3.

Note that the answer must be a **substring**, "pwke" is a *subsequence* and not a substring.

akuna OA

There are a number of products that Acme Corp. needs to have produced. For each product there is a worst case and an expected cost. Before starting a project, Acme must have at least enough cash on hand to pay the worst cost. Products can be produced in any order. If every product is produced at expected cost, determine the minimum beginning cash requirement to get all products produced.

Example:

worstCase = [6, 5, 7]

expected = [4, 2, 1]

The optimal order of production is product 2, then 1, then 0. If Acme starts with 9 units of cash on hand, they will have enough to start product 2 ($9 \geq 7$), and it will cost them 1 in the end. They will have $9 - 1 = 8$ cash on hand and be able to produce product 1 ($8 \geq 5$). It will cost them 2 in the end. They will have $8 - 2 = 6$ on hand and can start product 0 ($6 \geq 6$) and will pay 4 in the end. They will have $6 - 4 = 2$ units cash on hand at the end. The minimum starting amount is 9.

$1 \leq n \leq 10^5$ $1 \leq \text{expected}[i] \leq \text{worstCase}[i] \leq 10^5$

03题目讲解

HW 1

leetcode1137

1137. N-th Tribonacci Number

Easy

👍 66

💬 13

♡ Favorite

🔗 Share

The Tribonacci sequence T_n is defined as follows:

$T_0 = 0$, $T_1 = 1$, $T_2 = 1$, and $T_{n+3} = T_n + T_{n+1} + T_{n+2}$ for $n \geq 0$.

Given n , return the value of T_n .

Example 1:

Input: $n = 4$

Output: 4

Explanation:

$T_3 = 0 + 1 + 1 = 2$

$T_4 = 1 + 1 + 2 = 4$

Example 2:

Input: $n = 25$

Output: 1389537

Constraints:

- $0 \leq n \leq 37$
- The answer is guaranteed to fit within a 32-bit integer, ie. $\text{answer} \leq 2^{31} - 1$.

HW 2

leetcode46

46. Permutations

Medium

👍 2431

💬 76

♡ Favorite

🔗 Share

Given a collection of **distinct** integers, return all possible permutations.

Example:

Input: [1,2,3]

Output:

```
[  
  [1,2,3],  
  [1,3,2],  
  [2,1,3],  
  [2,3,1],  
  [3,1,2],  
  [3,2,1]  
]
```

HW 3

leetcode39

39. Combination Sum

Medium  2436  74  Favorite  Share

Given a **set** of candidate numbers (`candidates`) (**without duplicates**) and a target number (`target`), find all unique combinations in `candidates` where the candidate numbers sums to `target` .

The **same** repeated number may be chosen from `candidates` unlimited number of times.

Note:

- All numbers (including `target`) will be positive integers.
- The solution set must not contain duplicate combinations.

Example 1:

```
Input: candidates = [2,3,6,7], target = 7,  
A solution set is:  
[  
  [7],  
  [2,2,3]  
]
```

Example 2:

```
Input: candidates = [2,3,5], target = 8,  
A solution set is:  
[  
  [2,2,2,2],  
  [2,3,3],  
  [3,5]  
]
```

HW 4

leetcode77

77. Combinations

Medium

👍 929

💬 52

♡ Favorite

🔗 Share

Given two integers n and k , return all possible combinations of k numbers out of $1 \dots n$.

Example:

Input: $n = 4, k = 2$

Output:

```
[  
  [2,4],  
  [3,4],  
  [2,3],  
  [1,2],  
  [1,3],  
  [1,4],  
]
```