

assignmentB

Assignment #B : dp

Updated 1448 GMT+8 Nov 18, 2025

2025 fall, Compiled by 顾桂榕 基础医学院



顾桂榕 医学预科办

说明:

- 1) 请把每个题目解题思路 (可选), 源码 *Python*, 或者 *C++* (已经在 *Codeforces/Openjudge* 上 *AC*), 截图 (包含 *Accepted*), 填写到下面作业模版中 (推荐使用 *typora* <https://typoraio.cn>, 或者用 *word*)。 *AC* 或者没有 *AC*, 都请标上每个题目大致花费时间。
- 2) 提交时候先提交 *pdf* 文件, 再把 *md* 或者 *doc* 文件上传到右侧 “作业评论”。 *Canvas* 需要有同学清晰头像、提交文件有 *pdf*、 “作业评论” 区有上传的 *md* 或者 *doc* 附件。
- 3) 如果不能在截止前提交作业, 请写明原因。

1. 题目

LuoguP1255 数楼梯

dp, *bfs*, <https://www.luogu.com.cn/problem/P1255>

思路:

代码:

```

from functools import lru_cache
@lru_cache(maxsize=None)
def ways(n):
    dp = [0]*(n+1)
    dp[0]=1;dp[1]=1
    for i in range(2,n+1):
        dp[i]=dp[i-1]+dp[i-2]
    return dp
ways_list = ways(5000)
print(ways_list[int(input())])

```

代码运行截图 (至少包含有"*Accepted*")

R248014223 记录详情

编程语言

Python 3

代码地址

233

测试点信息

源代码

测试点信息

#1 AC 25ms/9.81MB	#2 AC 23ms/5.11MB	#3 AC 26ms/11.49MB	#4 AC 27ms/11.47MB	#5 AC 23ms/6.21MB	#6 AC 21ms/5.29MB	#7 AC 23ms/6.85MB
#8 AC 24ms/10.81MB	#9 AC 22ms/5.18MB	#10 AC 24ms/6.83MB				

Rosie211

所属题目

P1255 数楼梯

评测状态

Accepted

提交时间

2025-11-18 17:37:16

个人设置

我的收藏

我的团队

我的比赛

我的专栏

我的题库

锁定

登出

27528: 跳台阶

dp, <http://cs101.openjudge.cn/practice/27528/>

思路:

代码:

```

from functools import lru_cache
@lru_cache(maxsize=None)
def ways(n):
    dp = [0] * (n+1)
    dp[0]=1
    for _ in range(1,n+1):

```

```

        dp[_]=sum(dp[x] for x in range(0,_))
    return dp
ways_list = ways(25)
N = int(input())
print(ways_list[N])

```

代码运行截图 (至少包含有"Accepted")

#50892605提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

from functools import lru_cache
@lru_cache(maxsize=None)
def ways(n):
    dp = [0] * (n+1)
    dp[0]=1
    for _ in range(1,n+1):
        dp[_]=sum(dp[x] for x in range(0,_))
    return dp
ways_list = ways(25)
N = int(input())
print(ways_list[N])

```

基本信息

#: 50892605
 题目: 27528
 提交人: R.
 内存: 3612kB
 时间: 22ms
 语言: Python3
 提交时间: 2025-11-18 16:39:05

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

M23421: 《算法图解》小偷背包问题

dp, <http://cs101.openjudge.cn/pctbook/M23421/>

思路:

代码:

```

N,B = map(int,input().split())
price_list = list(map(int,input().split()))
weight_list = list(map(int,input().split()))
dp = [[0]*(B+1) for _ in range(N+1)]
for x in range(1,N+1):
    for y in range(1,B+1):
        if y>=weight_list[x-1]:
            dp[x][y]=max(dp[x-1][y],price_list[x-1]+dp[x-1][y-
weight_list[x-1]])
print(dp[-1][-1])

```

代码运行截图 (至少包含有"Accepted")

#50893691提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
N,B = map(int,input().split())
price_list = list(map(int,input().split()))
weight_list = list(map(int,input().split()))
dp = [[0]*(B+1) for _ in range(N+1)]
for x in range(1,N+1):
    for y in range(1,B+1):
        if y>=weight_list[x-1]:
            dp[x][y]=max(dp[x-1][y],price_list[x-1]+dp[x-1][y-weight_list[x-1]])
        else:
            dp[x][y]=dp[x-1][y]
print(dp[-1][-1])
```

基本信息

#: 50893691
题目: M23421
提交人: R.
内存: 3612kB
时间: 24ms
语言: Python3
提交时间: 2025-11-18 17:25:39

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

M5.最长回文子串

dp, two pointers, string, <https://leetcode.cn/problems/longest-palindromic-substring/>

思路:

代码:

```
s = input()
n = len(s)
max_len = 1
begin = 0
if n==1:
    print(s)
dp = [[False]*n for _ in range(n)]
for i in range(n):
    dp[i][i]=True
for interval in range(1,n):
    for L in range(n):
        R = L+interval
        if R >= n:
            break
        if s[L]!=s[R]:
            dp[L][R]=False
        else:
            if interval<3:
                dp[L][R]=True
            else:
                dp[L][R]=dp[L+1][R-1]
```

```

        if dp[L][R] and interval >= max_len:
            max_len = interval
            begin = L

print(s[begin:begin+max_len+1])

```

代码运行截图 (至少包含有 "Accepted")

通过 142 / 142 个通过的测试用例

提交于 2025.11.23 10:25

官方题解

写题解



面向在校学生的专享特惠

完成认证享 7 折 Plus 会员，享受更多学业及职业成长帮助



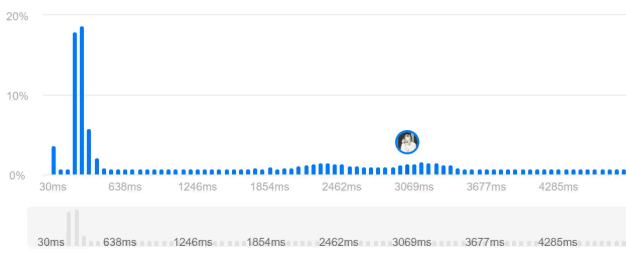
执行用时分布

消耗内存分布

3059 ms | 击败 21.63%

25.13 MB | 击败 10.52%

复杂度分析



代码 | Python3

```

1 class Solution:
2     def longestPalindrome(self, s: str) -> str:
3         n = len(s)
4         if n < 2:
5             return s
6
7         max_len = 1
8         begin = 0
9
10        dp = [[False] * n for _ in range(n)]
11        for i in range(n):
12            dp[i][i] = True
13
14        for L in range(2, n + 1):
15            for i in range(n):
16                j = L + i - 1
17
18                if j >= n:
19                    break
20
21                if s[i] != s[j]:
22                    dp[i][j] = False
23                else:
24                    if j - i < 3:
25                        dp[i][j] = True
26                    else:
27                        dp[i][j] = dp[i + 1][j - 1]
28
29                if dp[i][j] and j - i + 1 > max_len:
30                    max_len = j - i + 1
31                    begin = i
32        return s[begin:begin + max_len]

```

474D. Flowers

dp, 1700 <https://codeforces.com/problemset/problem/474/D>

思路:

- For $K = 2$ and length 1 Marmot can eat (R).
- For $K = 2$ and length 2 Marmot can eat (RR) and (WW).
- For $K = 2$ and length 3 Marmot can eat (RRR), (RWW) and (WWR).
- For $K = 2$ and length 4 Marmot can eat (WWWW), (RWWR), (WRRR), (RRWW), (RRRR).
- For $K = 2$ and length 5 Marmot can eat (WWWWR), (RWWWR), (RWWRR), (WRRRR), (RRRWW), (RRRRR).

代码:

```

MOD = 10**9+7
def solve():
    t, k = map(int, input().split())
    a_b_list = []
    max_b = 0
    for _ in range(t):
        a,b = map(int,input().split())
        a_b_list.append((a,b))
        max_b = max(max_b,b)

    if k>max_b:
        res = []
        for a,b in a_b_list:
            res.append(str((b-a+1)%MOD))
        print("\n".join(res))
        return

    dp = [0]*(max_b+1)
    for i in range(max_b+1):
        if i==0:
            dp[0]=1
        elif i<k:
            dp[i]=dp[i-1]
        else:
            dp[i]=(dp[i-k]+dp[i-1])%MOD

    prefix = [0]*(max_b+1)
    prefix[0]=dp[0]
    for i in range(1,max_b+1):
        prefix[i]=(prefix[i-1]+dp[i])%MOD
    res = []
    for a,b in a_b_list:
        res.append(str((prefix[b] - prefix[a - 1]) % MOD))
    print("\n".join(res))
    return

solve()

```

代码运行截图 (至少包含有"Accepted")

By Rose-Rong, contest: Codeforces Round 271 (Div. 2), problem: (D) Flowers, **Accepted**, #, [Copy](#).

```
MOD = 10**9+7
def solve():
    t, k = map(int, input().split())
    a_b_list = []
    max_b = 0
    for _ in range(t):
        a, b = map(int, input().split())
        a_b_list.append((a, b))
        max_b = max(max_b, b)

    if k > max_b:
        res = []
        for a, b in a_b_list:
            res.append(str((b-a+1)%MOD))
        print("\n".join(res))
        return

    dp = [0]*(max_b+1)
    for i in range(max_b+1):
        if i==0:
            dp[0]=1
        elif i<k:
            dp[i]=dp[i-1]
        else:
            dp[i]=(dp[i-k]+dp[i-1])%MOD

    prefix = [0]*(max_b+1)
    prefix[0]=dp[0]
    for i in range(1, max_b+1):
        prefix[i]=(prefix[i-1]+dp[i])%MOD
    res = []
    for a, b in a_b_list:
        res.append(str((prefix[b] - prefix[a - 1]) % MOD))
    print("\n".join(res))
    return

solve()
```

M198.打家劫舍

dp, <https://leetcode.cn/problems/house-robber/>

思路:

代码:

```
class Solution:

    def rob(self, nums: List[int]) -> int:

        def money_max(n):

            dp = [0]*len(nums)

            dp[0]=nums[0]

            dp[1]=max(nums[0], nums[1])

            for _ in range(2, n):
```

```

        dp[_]=max(dp[_-1],dp[_-2]+nums[_])

    return dp[n-1]

    if len(nums)<3:

        return max(nums)

    else:

        return money_max(len(nums))

```

代码运行截图 (至少包含有"Accepted")

通过 70 / 70 个通过的测试用例 用时: 8 hrs 15 m 59 s

宙 提交于 2025.11.23 11:22

官方题解

写题解



面向在校学生的专享特惠

完成认证享 7 折 Plus 会员, 享受更多学业及职业成长帮助



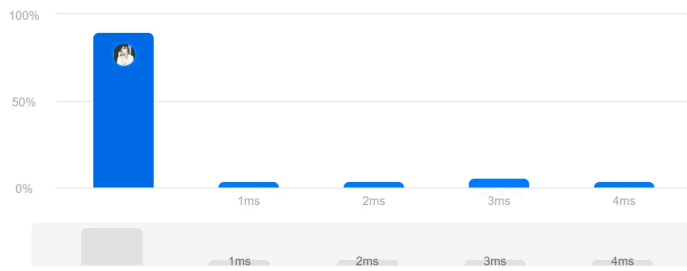
① 执行用时分布

0 ms | 击败 100.00%

复杂度分析

② 消耗内存分布

17.69 MB | 击败 13.60%



```

1 class Solution:
2     def rob(self, nums: List[int]) -> int:
3         def money_max(n):
4             dp = [0]*len(nums)
5             dp[0]=nums[0]
6             dp[1]=max(nums[0],nums[1])
7             for _ in range(2,n):
8                 dp[_]=max(dp[_-1],dp[_-2]+nums[_])
9             return dp[n-1]
10
11         if len(nums)<3:
12             return max(nums)
13         else:
14             return money_max(len(nums))

```

已存储

2. 学习总结和收获

如果作业题目简单, 有否额外练习题目, 比如: OJ“计概2024fall每日选做”、CF、LeetCode、洛谷等网站题目。

感觉dp题目的状态转移方程的寻找是很重要的。

一开始没什么思路

现在觉得方程其实都和斐波那契长得很像

其实就是寻找(n)和(n-1)&(n-2)之间的数量关系

有的时候需要分类讨论

虽然现在寻找方程的耗时较长, 但是我想多积累就会有灵感了

*FLOWERS*这一题先是超内存，然后超时
问了 *AI*

1.寻找 max_b ，算到头后用前缀和直接调用结果

前缀和：算第 i 项到第 k 项的和（不用直接 *sum* 进行重复运算）

2.能改进 $max_b < k$ 的情况

另积累一下写法：

```
def solve():
```

内部写 *input*, 再进行处理

将结果添加至列表(注意要转化为 *str* 字符串)

```
print("\n".join(res))
```

```
return
```

调用函数 *solve()*