

# Assignment #B: dp

Updated 1448 GMT+8 Nov 18, 2025

2025 fall, Complied by 贺桢羽, 心理与认知科学学院

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：斐波拉契数列

代码：

```
a=0
b=1
n=int(input())
for i in range(n):
    a,b=b,a+b
print(b)
```

代码运行截图 (至少包含有"Accepted")

The screenshot shows the Luogu platform interface for problem P1255. At the top, there is a summary bar with programming language (Python 3), code length (64B), runtime (215ms), and memory usage (10.39MB). Below this, the user information is shown: kuku0123, belonging to P1255 数楼梯, status Accepted, score 100, and submission time 2025-10-20 21:33:06. On the left, there is a 'Test Point Information' section with two rows of ten test points each, all marked as Accepted (AC) with their respective runtimes and memory usage. The first row includes test points #1 through #7, and the second row includes #8 through #10. The right side of the screenshot shows the detailed submission statistics.

所属题目	P1255 数楼梯
评测状态	Accepted
评测分数	100
提交时间	2025-10-20 21:33:06

## 27528: 跳台阶

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

思路: dp

代码:

```
n=int(input())
dp=[0 for _ in range(n+1)]
if n>=3:
    dp[1]=1
    dp[2]=2
    for i in range(3,n+1):
        dp[i]=sum(dp[:i])+1
    print(dp[n])
if n==2:
    print(2)
if n==1:
    print(1)
```

代码运行截图 (至少包含有"Accepted")

#50892902提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 50892902  
题目: 27528  
提交人: 25n2500013720  
内存: 3620kB  
时间: 22ms  
语言: Python3  
提交时间: 2025-11-18 16:51:30

源代码

```
n=int(input())
dp=[0 for _ in range(n+1)]
if n>=3:
    dp[1]=1
    dp[2]=2
    for i in range(3,n+1):
        dp[i]=sum(dp[:i])+1
    print(dp[n])
if n==2:
    print(2)
if n==1:
    print(1)
```

©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: 倒着分析, 每次计算都是在没有加这一钟物品最优情况下, 再去寻找最优解。

代码:

```

n,b=map(int,input().split())
pri=list(map(int,input().split()))
wei=list(map(int,input().split()))
dp=[0 for _ in range(b+1)]
for s in range(n):
    w=wei[s]
    p=pri[s]
    for i in range(b,w-1,-1):
        dp[i]=max(dp[i],dp[i-w]+p)
print(dp[b])

```

代码运行截图 (至少包含有"Accepted")

#50894407提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 50894407  
题目: M23421  
提交人: 25n2500013720  
内存: 3592kB  
时间: 22ms  
语言: Python3  
提交时间: 2025-11-18 18:28:02

源代码

```

n,b=map(int,input().split())
pri=list(map(int,input().split()))
wei=list(map(int,input().split()))
dp=[0 for _ in range(b+1)]
for s in range(n):
    w=wei[s]
    p=pri[s]
    for i in range(b,w-1,-1):
        dp[i]=max(dp[i],dp[i-w]+p)
print(dp[b])

```

©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：双指针。

代码：

```

class Solution:
    def longestPalindrome(self, s: str) -> str:
        n=len(s)
        if n==1:
            return str(s)
        if n==2:
            if s[0]==s[1]:
                return s
            else:
                return s[0]
        else:
            temp=0
            ans=s[0]
            for i in range(n-1):
                check=0
                a=float("-inf")
                b=float("-inf")
                if i+2<=n-1 and s[i]==s[i+2]:
                    a=i

```

```

        b=i+2
        check=1
    while a>=1 and b<=n-2 and check:
        a-=1
        b+=1
        if s[a]!=s[b]:
            a+=1
            b-=1
            break
    if b-a+1>temp:
        temp=b-a+1
        ans=s[a:b+1]
    if s[i]==s[i+1]:
        a=i
        b=i+1
        check=1
    while a>=1 and b<=n-2 and check:
        a-=1
        b+=1
        if s[a]!=s[b]:
            a+=1
            b-=1
            break
    if b-a+1>temp:
        temp=b-a+1
        ans=s[a:b+1]
return ans

```

### 代码运行截图 (至少包含有"Accepted")



通过 142 / 142 个通过的测试用例 用时: 52 ms 17 s  
荷莲鱼 提交于 2025.11.18 23:08

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 ② 消耗内存分布  
267 ms | 击败 68.38% 17.48 MB | 击败 88.52%

复杂度分析

Python3 智能模式

```

1 class Solution:
2     def longestPalindrome(self, s: str) -> str:
3         n=len(s)
4         if n==1:
5             return str(s)
6         if n==2:
7             if s[0]==s[1]:
8                 return s
9             else:
10                return s[0]
11 else:
12     temp=0
13     ans=s[0]
14     for i in range(n-1):
15         check=0
16         a=float("-inf")
17         b=float("-inf")
18         if i+2>n-1 and s[i]==s[i+2]:
19             a=i
20             b=i+2
21             check=1
22         while a>=1 and b<=n-2 and check:
23             a-=1
24             b+=1
25             if s[a]==s[b]:
26                 a+=1
27                 b-=1
28                 break
29         if b-a+1>temp:
30             temp=b-a+1
31             ans=s[a:b+1]
32             if s[i]==s[i+1]:
33                 a=i
34                 b=i+1
35                 check=1
36             while a>=1 and b<=n-2 and check:
37                 a-=1
38                 b+=1
39                 if s[a]==s[b]:
40                     a+=1

```

已存储 行 31, 列 33

## 474D. Flowers

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

思路：最开始一直超时，最终使用预处理+前缀和处理才通过。

代码：

```
t,k=map(int,input().split())
dp = [0] * (100000 + 1)
dp[0] = 1
dp[1] = 1
for i in range(1, 100000+ 1):
    if i - k >= 0:
        dp[i] = (dp[i - 1] + dp[i - k]) % (1000000007)
    else:
        dp[i] = (dp[i - 1])
sum=[0] * (100000 + 1)
for p in range(1,100000+ 1):
    sum[p]=(dp[p]+sum[p-1])% (1000000007)
for _ in range(t):
    a, b = map(int, input().split())
    print((sum[b]-sum[a-1])% (1000000007))
```

代码运行截图 (至少包含有"Accepted")

By hzy499, contest: Codeforces Round 271 (Div. 2), problem: (D) Flowers, Accepted, #, Copy.

```
t,k=map(int,input().split())
dp = [0] * (100000 + 1)
dp[0] = 1
dp[1] = 1
for i in range(1, 100000+ 1):
    if i - k >= 0:
        dp[i] = (dp[i - 1] + dp[i - k]) % (1000000007)
    else:
        dp[i] = (dp[i - 1])
sum=[0] * (100000 + 1)
for p in range(1,100000+ 1):
    sum[p]=(dp[p]+sum[p-1])% (1000000007)
for _ in range(t):
    a, b = map(int, input().split())
    print((sum[b]-sum[a-1])% (1000000007))
```

### →Judgement Protocol

Test: #1, time: 124 ms., memory: 7728 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```
3 2
1 3
2 3
4 4
```

Output

```
6
```

```
5
```

```
5
```

Answer

```
6
```

```
5
```

```
5
```

Checker Log

```
ok 3 number(s): "6 5 5"
```

Test: #2, time: 93 ms., memory: 4680 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```
139 99252
70888 79845
2746 73795
86884 88479
26655 17050
```

## M198. 打家劫舍

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

思路：注意边界的处理。

代码：

```
class Solution:
    def rob(self, nums: List[int]) -> int:
        dp=[0]*len(nums)
        dp[0]=nums[0]
        for i in range(1,len(nums)):
            temp=dp[i-1]
            if i-1>0:
                for x in range(i-1):
                    if dp[x]+nums[i]>temp:
                        temp=dp[x]+nums[i]
            dp[i]=temp
            if i-1==0:
                if nums[1]>nums[0]:
                    dp[i]=nums[1]
                else:
                    dp[i]=nums[0]
        return dp[len(nums)-1]
```

### 代码运行截图 (至少包含有"Accepted")

The screenshot shows a LeetCode submission page for the 'House Robber' problem. The top bar includes links for '题目描述', '通过', '题解', '提交记录'. The main area displays the following information:

- 通过**: 70 / 70 个通过的测试用例 用时: 10 ms 52 s
- 提交于 2025.11.19 16:47
- 面向在校学生的专享特惠**: 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**: 0 ms | 击败 100.00%  
复杂度分析
- 消耗内存分布**: 17.64 MB | 击败 25.84%
- 代码** (Python3):

```
1 class Solution:
2     def rob(self, nums: List[int]) -> int:
3         dp=[0]*len(nums)
4         dp[0]=nums[0]
5         for i in range(1,len(nums)):
6             temp=dp[i-1]
7             if i-1>0:
8                 for x in range(i-1):
9                     if dp[x]+nums[i]>temp:
10                         temp=dp[x]+nums[i]
11             dp[i]=temp
12             if i-1==0:
13                 if nums[1]>nums[0]:
14                     dp[i]=nums[1]
15                 else:
16                     dp[i]=nums[0]
17         return dp[len(nums)-1]
```

## 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。

这周学习dp，个人感觉比dfs好理解很多，只是需要一个精巧的数学递推公式构造就可以很简单地做出来。然后，做了这么多dp题，发现最开始递推那几项一定要谨慎，很多时候都是边界条件没有考虑导致WA和RE。然后在做Flower这个题时又使用了数组前缀和进行优化，这次是一维数组，上次做垃圾炸弹那道题目时用到的是二维数组前缀和。最长回文子串没有使用dp，感觉直接使用双指针也可以思路比较清晰地解决。在处理背包问题时，发现一定要考虑dp从前往后和从后往前解决，不然会出错。

另外，我发现了老师布置在其他网站的题目可以清楚地看到测试数据，这样也很方便就能debug出来了，但是OJ上的题只要以RE或者WA就只能自己一点一点地debug，要是考试的时候有测试数据就好了呜呜。