Assignment #B: dp

Updated 1448 GMT+8 Nov 18, 2025

2025 fall, Complied by 郑文萱 基础医学院

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

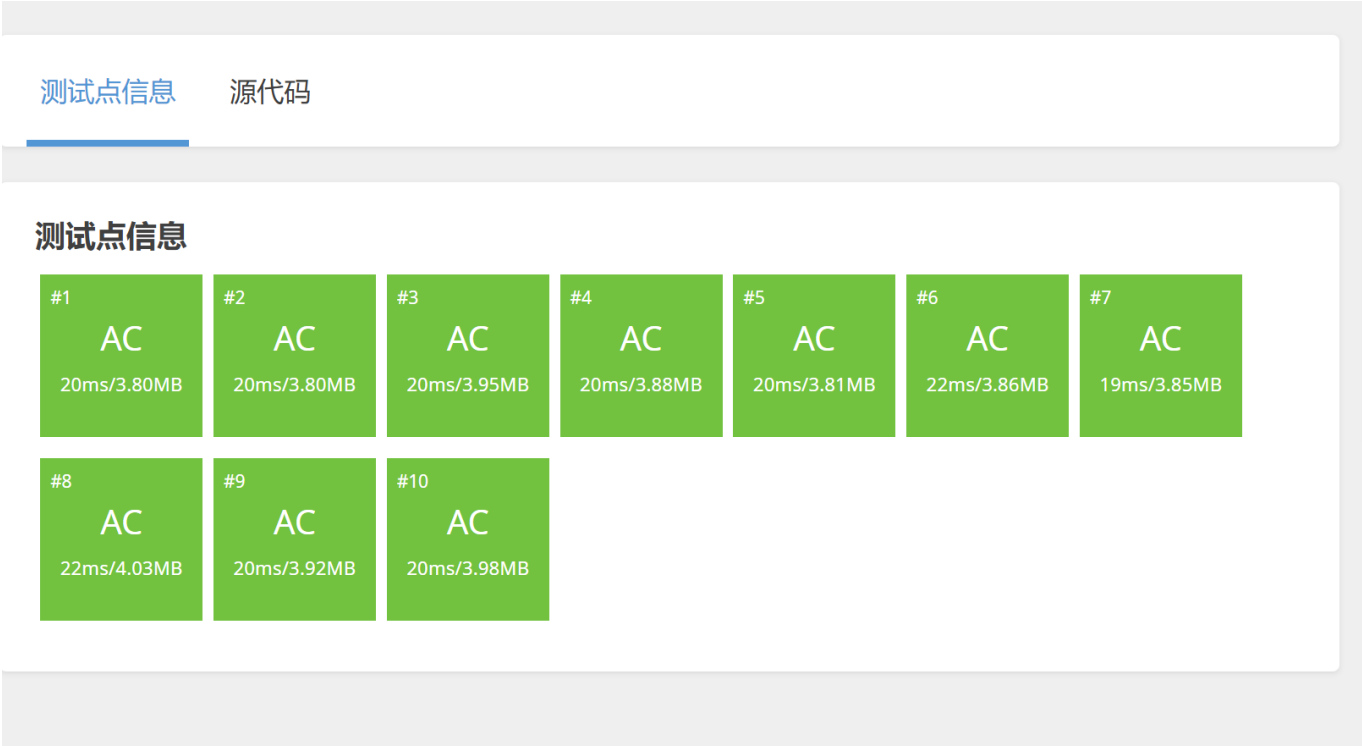
思路：a1=1,a2=1,an=an-1+an-2,其实就是斐波那契数列。第一次把迭代的方法提交，非常完美；第二次交了无脑递归的做法，爆栈+超时；第三次dp，顺利通过。对于此题来说，迭代是最优的做法，因为只储存前两个值节省空间，但如果输入有多个，dp是毫无疑问的最优。

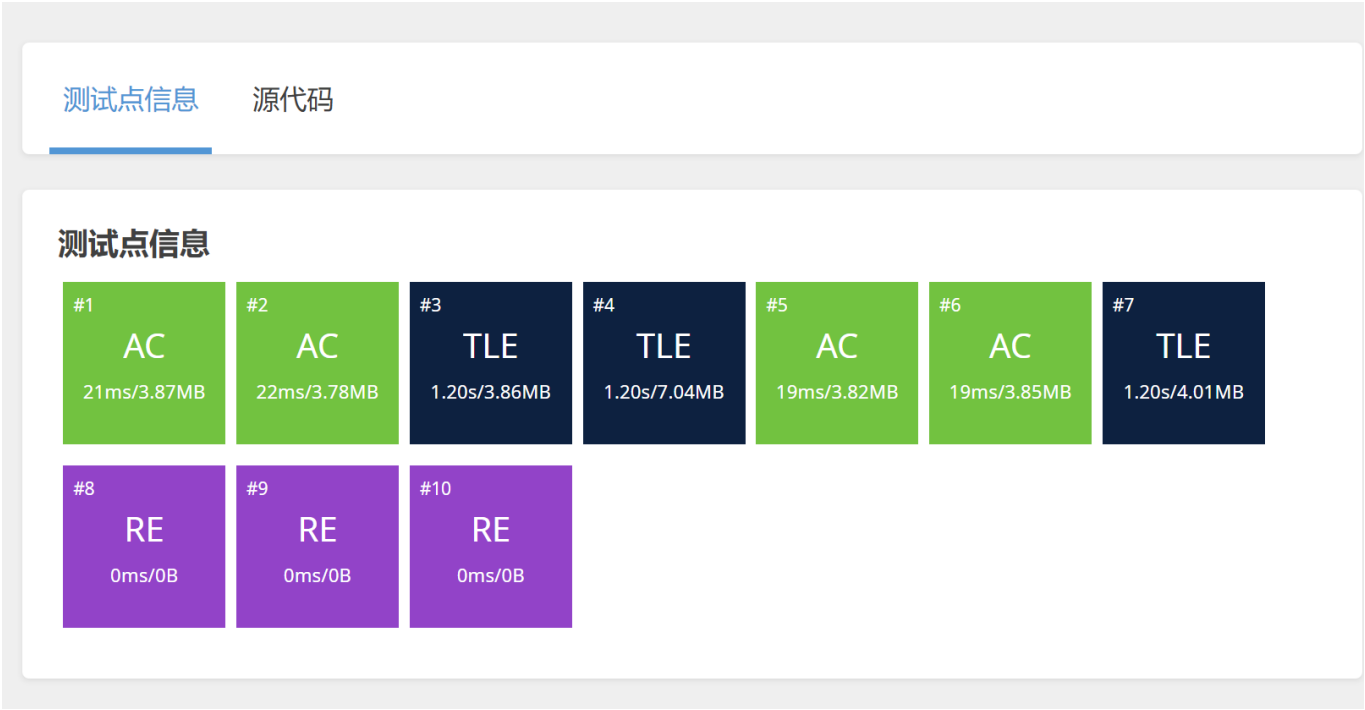
代码：

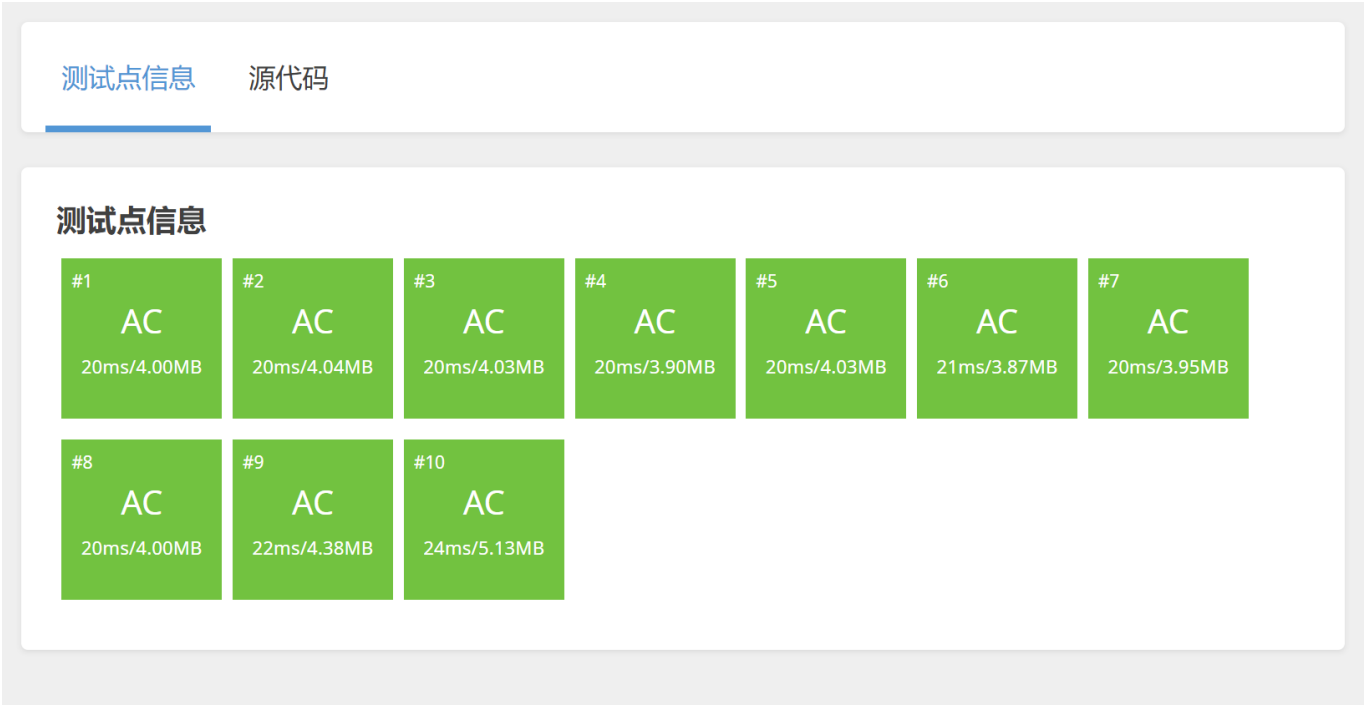
```
#第一次测试
def fib(n):
    if n == 1 or n == 2:
        return 1
    a, b = 1, 1
    for _ in range(n - 2):
        a, b = b, a + b
    return b
t = int(input())
print(fib(t+1))

#第二次测试
n=int(input())
def fib(n):
    if n==1 or n==2:
        return 1
    else:
        return fib(n-1)+fib(n-2)
print(fib(n+1))

#第三次测试
```

```python
n = int(input())
def fib_dp(n):
    if n == 1 or n == 2:
        return 1
    dp = [0] * (n + 1)
    dp[1] = 1
    dp[2] = 1
    for i in range(3, n + 1):
        dp[i] = dp[i - 1] + dp[i - 2]
    return dp[n]
print(fib_dp(n + 1))
```

代码运行截图 （至少包含有"Accepted"）







27528: 跳台阶

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

思路：此题意义不明： f(N) = f(N-1) + f(N-2) + ... + f(1) + f(0)；f(0) = 1，那么f(N)=2^(n-1)。 但标签是dp而不是math之类的，只能说这是个递推式太明显能推出最终表达式的dp，而代码只能说是非常简洁了（）。

代码：

```
n=int(input())
print(2**(n-1))
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
n = int(input())
print(2 ** (n - 1))
```

M23421:《算法图解》小偷背包问题

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

思路：看了算法图解上背包问题这一章节，很有意思，讲得好清楚，已入坑此书，希望下次交作业前能看一半！ 此题的递推式原理：设最大价值为dp[i][j]，若不拿第i件商品，dp[i][j]=dp[i-1][j]；若拿，dp[i][j]=dp[i-1][j-weight[i]]+price[i]，取max。

代码：

```
def backpack(n,b,price,weight):
    dp=[[0]*(b+1) for i in range(n+1)]
    for i in range(1,n+1):
        for j in range(1,b+1):
            dp[i][j]=dp[i-1][j]
            if j>=weight[i-1]:
                dp[i][j]=max(dp[i][j],dp[i-1][j-weight[i-1]]+price[i-1])
    return dp[n][b]
n,b=map(int,input().split())
price=list(map(int,input().split()))
weight=list(map(int,input().split()))
print(backpack(n,b,price,weight))
```

代码运行截图 （至少包含有"Accepted"）

状态：Accepted

源代码

```
def backpack(n,b,price,weight):
    dp=[[0]*(b+1) for i in range(n+1)]
    for i in range(1,n+1):
        for j in range(1,b+1):
            dp[i][j]=dp[i-1][j]
            if j>=weight[i-1]:
                dp[i][j]=max(dp[i][j],dp[i-1][j-weight[i-1]]+price[i-1]
    return dp[n][b]
n,b=map(int,input().split())
price=list(map(int,input().split()))
weight=list(map(int,input().split()))
print(backpack(n,b,price,weight))
```


M5.最长回文子串

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

思路：此题标签既可以用双指针也可以用dp，但双指针还是简单一点，幸好样例给了奇数和偶数的回文子串，不然我真的可能会发现不了，以及要注意索引取值的问题。

代码：

```
class Solution:
    def longestPalindrome(self, s: str) -> str:
        def find(l,r):
            while l>=0 and r<len(s) and s[l]==s[r]:
                l-=1
                r+=1
            return s[l+1:r]
        string=''
        for i in range(len(s)):
            odd=find(i,i)
            even=find(i,i+1)
            if len(odd)>len(string):
                string=odd
            if len(even)>len(string):
                string=even
        return string
```

代码运行截图 （至少包含有"Accepted"）

474D. Flowers

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

思路：这个题一开始没看懂（翻译太抽象了），让DS翻译之后懂了。递推式倒不太难：i>=k时可放白色，则f[i]=f[i-1]（最后为红色）+f[i-k]（最后k个为白色）；i < k时不可放白色，则f[i]=f[i-1]（全是红色）。 在此题题解中学习了前缀和降低时间复杂度的做法，以及了解了计算机竞赛中大数常常要求取模的特点还有一些取模的技巧（比如额外加上MOD以避免负数）。可恶啊，1e9竟然是浮点数!

代码：

```
mod=1000000007
dp=[0]*100001
dp[0]=1
sum_dp=[0]*100001
```

```
t,k=map(int,input().split())
for i in range(1,100001):
    dp[i]=dp[i-1]%mod
    if i>=k:
        dp[i]=(dp[i-1]+dp[i-k])%mod
    sum_dp[i]=(sum_dp[i-1]+dp[i])%mod
for _ in range(t):
    m,n=map(int,input().split())
    print((sum_dp[n]-sum_dp[m-1]+mod)%mod)
```

代码运行截图 （至少包含有"Accepted"）

350020812	Nov/21/2025 13:32UTC+8	zwx0208	D - Flowers	Python 3	Accepted	733 ms	7800 KB

M198.打家劫舍

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

思路：这个太简单，状态转移方程很好想：上次偷了，这次不能偷；上次没偷，这次必偷。所以就是dp[i]=max(dp[i-1],dp[i-2]+nums[i])

代码：

```
class Solution:
    def rob(self, nums: List[int]) -> int:
        if not nums:
            return 0
        n=len(nums)
        if n==1:
            return nums[0]
        dp=[0]*n
        dp[0]=nums[0]
        dp[1]=max(nums[0], nums[1])
        for i in range(2, n):
            dp[i]=max(dp[i-1],dp[i-2]+nums[i])
        return dp[n-1]
```

代码运行截图 （至少包含有"Accepted"）

代码 | Python3

```python
class Solution:
    def rob(self, nums: List[int]) -> int:
        if not nums:
            return 0
        n=len(nums)
        if n==1:
            return nums[0]
        dp=[0]*n
```

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。 这次作业练习的都是dp的基础题，很有收获（终于在做作业的时候有种巩固基础的感觉了）。本周计划是继续每日选做（还差一个月才能赶上进度有点难受了），以及读算法图解这本书。