# Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Complied by 同学的姓名、院系

**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

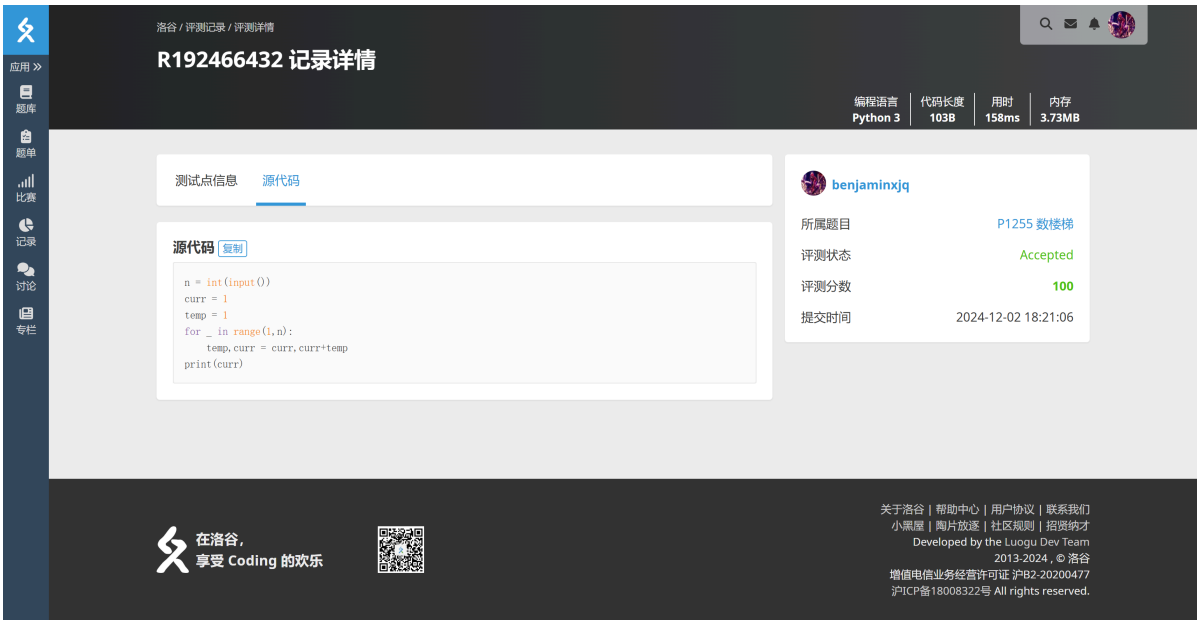
dp, bfs, https://www.luogu.com.cn/problem/P1255

思路：

代码：

```
n = int(input())
curr = 1
temp = 1
for _ in range(1,n):
    temp,curr = curr,curr+temp
print(curr)
```

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：

感觉不用dp做，小小偷个懒

代码：

```
n = int(input())
print(1 << n-1)
```

代码运行截图 （至少包含有"Accepted"）

# 474D. Flowers

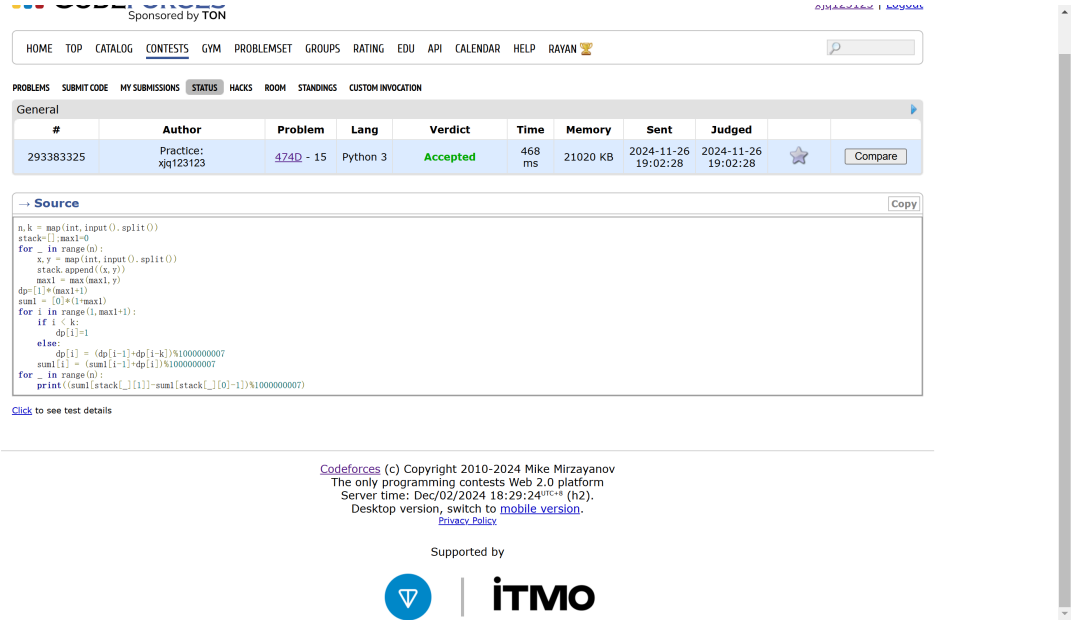
dp, https://codeforces.com/problemset/problem/474/D

思路：

代码：

```
n,k = map(int,input().split())
stack=[];max1=0
for _ in range(n):
    x,y = map(int,input().split())
    stack.append((x,y))
    max1 = max(max1,y)
dp=[1]*(max1+1)
sum1 = [0]*(1+max1)
for i in range(1,max1+1):
    if i < k:
        dp[i]=1
    else:
        dp[i] = (dp[i-1]+dp[i-k])%1000000007
    sum1[i] = (sum1[i-1]+dp[i])%1000000007
for _ in range(n):
    print((sum1[stack[_][1]]-sum1[stack[_][0]-1])%1000000007)
```

代码运行截图 （至少包含有"Accepted"）




# LeetCode5.最长回文子串

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

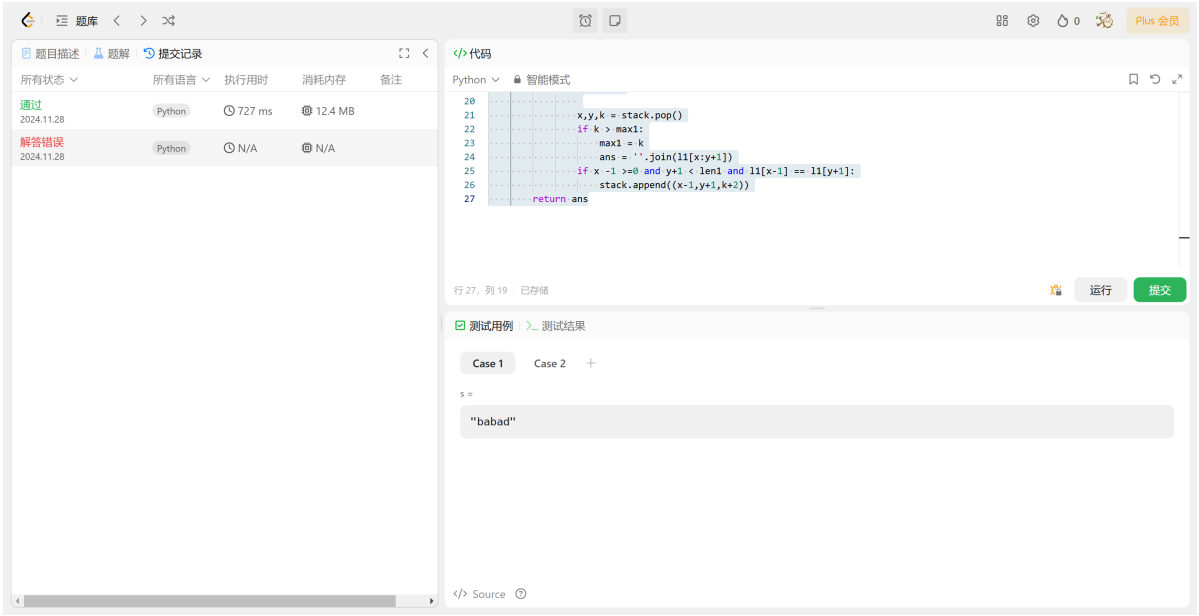
思路：

代码：

```
class Solution(object):
    def longestPalindrome(self, s):
        """
        :type s: str
        :rtype: str
        """
        l1 =s
        ans = l1[0]
        max1 = 0
        len1 = len(l1)
        for i in range(len1):
            stack = [(i,i,1)]
            step = 1
            while i+step < len1 and l1[i] == l1[i+step]:
                stack.pop()
                stack.append((i,i+step,1+step))
                step += 1
            i = i+step  -1
            while stack:

                x,y,k = stack.pop()
                if k > max1:
                    max1 = k
                    ans = ''.join(l1[x:y+1])
                if x -1 >=0 and y+1 < len1 and l1[x-1] == l1[y+1]:
                    stack.append((x-1,y+1,k+2))
        return ans
```

代码运行截图 （至少包含有"Accepted"）



## 12029: 水淹七军

bfs, dfs, http://cs101.openjudge.cn/practice/12029/

思路:

代码:

```
import sys
sys.setrecursionlimit(300000)
input = sys.stdin.read
data = input().split()
index1 = 1
K = int(data[0])
dx = [1,-1,0,0]
dy = [0,0,1,-1]
result = []
def check(mat,x,y,m,n,h):
    if 0<=x and x < m and 0<= y and y < n and mat[x][y] < h: return True
    return False
for _ in range(K):
    M ,N = map(int,data[index1:index1+2])
    index1 += 2
    matrix = [[] for _ in range(M)]
    flooded = [[False]*N for _ in range(M)]
    cases=[]
    for i in range(M):
        matrix[i]=list(map(int,data[index1:index1+N]))
        index1+=N
    idx,idy = map(int,data[index1:index1+2])
    index1+=2
    p = int(data[index1])
    index1+=1
    stat = False
    for i in range(p):
        x,y = map(int,data[index1:index1+2])
        index1+=2
        cases.append((x,y))
    inq = set()
    for a,b in cases:
        if stat:break
        stack=[(a-1,b-1)]
        # print(stack)
        # print(flooded)
        height = matrix[a-1][b-1]
        while stack:
            x,y= stack.pop()
            inq.add((x,y))
            if x == idx -1 and y == idy -1:
                stat = True
                break
            for i in range(4):
                nx = x + dx[i]
                ny = y + dy[i]
```

```
                if check(matrix,nx,ny,M,N,height) and (nx,ny) not in inq:
                    stack.append((nx,ny))

    result.append("Yes" if stat else "No")
sys.stdout.write("\n".join(result)+"\n")
```

代码运行截图 （至少包含有"Accepted"）


## 02802: 小游戏

bfs, http://cs101.openjudge.cn/practice/02802/

思路：

代码：

```
from collections import deque
M,N  = map(int,input().split())
test = 1
dx = [1,-1,0,0]
dy = [0,0,1,-1]
min1 = float("inf")

def check(x,y):
    if 0<=x and x<= M+1 and 0<= y and y <= N+1 and matrix[y][x] != 'X':return
True
    return False

while M:
    matrix = [[0]*(M+2) for _ in range(N+2)]
    visited = [[False]*(M+2) for _ in range(N+2)]
    for i in range(N):
        matrix[1+i][1:M+1] = list(input())
    x1,y1,x2,y2 = map(int,input().split())
```

```
    ans=[]
    l = 0
    # print(matrix)
    while x1+x2+y1+y2:
        matrix[y2][x2] =' '
        stack = deque()
        stack.append((x1,y1,-1,0))
        inq = set()
        inq.add((x1,y1))
        stat = False
        while stack:
            if stat:break
            x,y,dirc,k = stack.popleft()
            # print(x,y,k)
            if y == y2 and x == x2:
                    # print(nx,ny,k + int(i!=dirc))
                    ans.append(k)
                    stat = True
                    break
            # print(x,y,k)
            for i in range(4):
                nx = x+dx[i]
                ny = y+dy[i]
                while check(nx,ny) and (nx,ny) not in inq:
                    stack.append((nx,ny,i,k+1))
                    inq.add((nx,ny))
                    nx += dx[i]
                    ny += dy[i]
        if not stat:ans.append(0)
        matrix[y2][x2] ='X'
        x1,y1,x2,y2 = map(int,input().split())
        l += 1
    print(f"Board #{test}:")
    for i in range(l):
        if ans[i]:
            print(f"Pair {i+1}: {ans[i]} segments.")
        else:
            print(f"Pair {i+1}: impossible.")
    print()
    test += 1
    M,N  = map(int,input().split())
```

代码运行截图 （至少包含有"Accepted"）


# 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。

审题时注意要不要取模，两次都忘了

debug能力有提高，尽管还是比较耗时，因此有点担心机考。另：感觉自己有时候过于依赖群里的测试数据了，形成思维上的懒惰，不自己构造数据、去想corner case，感觉应当提高一下这方面能力，但也担心考试的时候根本没有那么多的时间去构造数据，请问老师有没有什么建议