

# Assignment #A: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by 颜鼎堃 工学院

## 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:

- 就是斐波那契数列
- 如果真要按照标的数据范围，下面代码是过不了的，因为递归深度不够

代码:

```
1  from functools import lru_cache
2  from sys import setrecursionlimit
3  setrecursionlimit(1 << 30)
4  @lru_cache
5  def Fibonacci(n):
6      return Fibonacci(n - 1) + Fibonacci(n - 2) if n > 2 else 1 if n == 1 else 2
7  print(Fibonacci(int(input())))
```

Python

应用 >>

题库

题单

比赛

记录

讨论

专栏

洛谷 / 评测记录 / 评测详情

R191222392 记录详情

编程语言Python 3 | 代码长度229B | 用时169ms | 内存7.54MB

测试点信息源代码

测试点信息

#1 AC 16ms/3.70MB	#2 AC 16ms/3.70MB	#3 AC 16ms/3.61MB	#4 AC 16ms/3.68MB	#5 AC 16ms/3.73MB	#6 AC 16ms/3.71MB	#7 AC 16ms/3.98MB
#8 AC 17ms/4.43MB	#9 AC 19ms/5.59MB	#10 AC 21ms/7.54MB				

ChaltinlenYan

所属题目P1255 数楼梯

评测状态Accepted

评测分数100

提交时间2024-11-26 15:23:05

在洛谷，  
享受 Coding 的欢乐

关于洛谷 | 帮助中心 | 用户协议 | 联系我们  
小黑屋 | 图片放逐 | 社区规则 | 招贤纳士  
Developed by the Luogu Dev Team  
2013-2024, © 洛谷  
增值电信业务经营许可证 沪B2-20200477  
沪ICP备18008322号 All rights reserved.

27528: 跳台阶

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

思路：

- 根据数学方法得到公式 $2^n - 1$

代码：

1

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

Python

## 代码运行截图 (至少包含有"Accepted")



## 474D. Flowers

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

思路:

- 先用 $k = 2$ 找思路, 根据排列组合公式发现又是斐波那契数列, 再合理推广找递推公式

$$a_n = a_{n-1} + a_{n-k} \text{ 就好}$$

代码:

```
1 from itertools import accumulate
2 t, k = map(int, input().split())
3 MOD = int(1e9 + 7)
4 m = int(1e5 + 1)
5 dp = [1 for i in range(m)]
6 for j in range(k, m):
7     dp[j] = (dp[j - 1] + dp[j - k]) % MOD
8 dp = tuple(accumulate(dp, func=lambda x, y: (x + y) % MOD))
9 for i in range(t):
10     a, b = map(int, input().split())
11     print((dp[b] - dp[a - 1]) % MOD)
```

Python

## 代码运行截图 (至少包含有"Accepted")

1860\*1101

CODEFORCES  
Sponsored by TON

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP RAYAN

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
293806345	Practice: Chaltinlen	474D - 15	Python 3	Accepted	1233 ms	9500 KB	2024-11-29 11:38:32	2024-11-29 11:38:32	★	Compare

→ Source Copy

```
from itertools import accumulate
t, k = map(int, input().split())
MOD = int(1e9 + 7)
a = int(1e5 + 1)
dp = [1 for i in range(a)]
for j in range(k, a):
    for i in range(j - 1, j - k):
        dp[j] = (dp[j - 1] + dp[j - k]) % MOD
dp = tuple(accumulate(dp, func=lambda x, y: (x + y) % MOD))
for i in range(t):
    a, b = map(int, input().split())
    print((dp[b] - dp[a - 1]) % MOD)
```

Click to see test details

Codeforces (c) Copyright 2010-2024 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Dec/01/2024 17:44:19 UTC+8 (13).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#)

Supported by

ITMO

## LeetCode5.最长回文子串

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

思路：

- 最开始我没看到题目要求子串必须连续！我想了很久，想到了可能要把原字符串逆序但不知道逆序之后干什么，然后一个同学告诉我直接求最长公共子序列就好，感觉瞬间明白了
- 然后发现子串要求连续，在原来程序的基础上，取出所有的公共子序列，再找其中既是回文的又是最长的那个，也算是过了

代码：

```
1 class Solution:
2     def longestPalindrome(self, s: str) -> str:
3         t = "".join(reversed(s))
4         n = len(s)
5         strings = ["" for i in range(n + 2)] for j in range(n + 2)]
6         for i in range(n):
7             for j in range(n):
8                 if s[i] == t[j]:
9                     strings[i + 1][j + 1] = strings[i][j] + s[i]
10        pos_pal = set()
11        max_par = s[0]
12        for i in map(set, strings):
13            pos_pal |= i
14        for i in pos_pal:
15            if i and i == i[::-1]:
16                max_par = max(max_par, i, key=len)
```

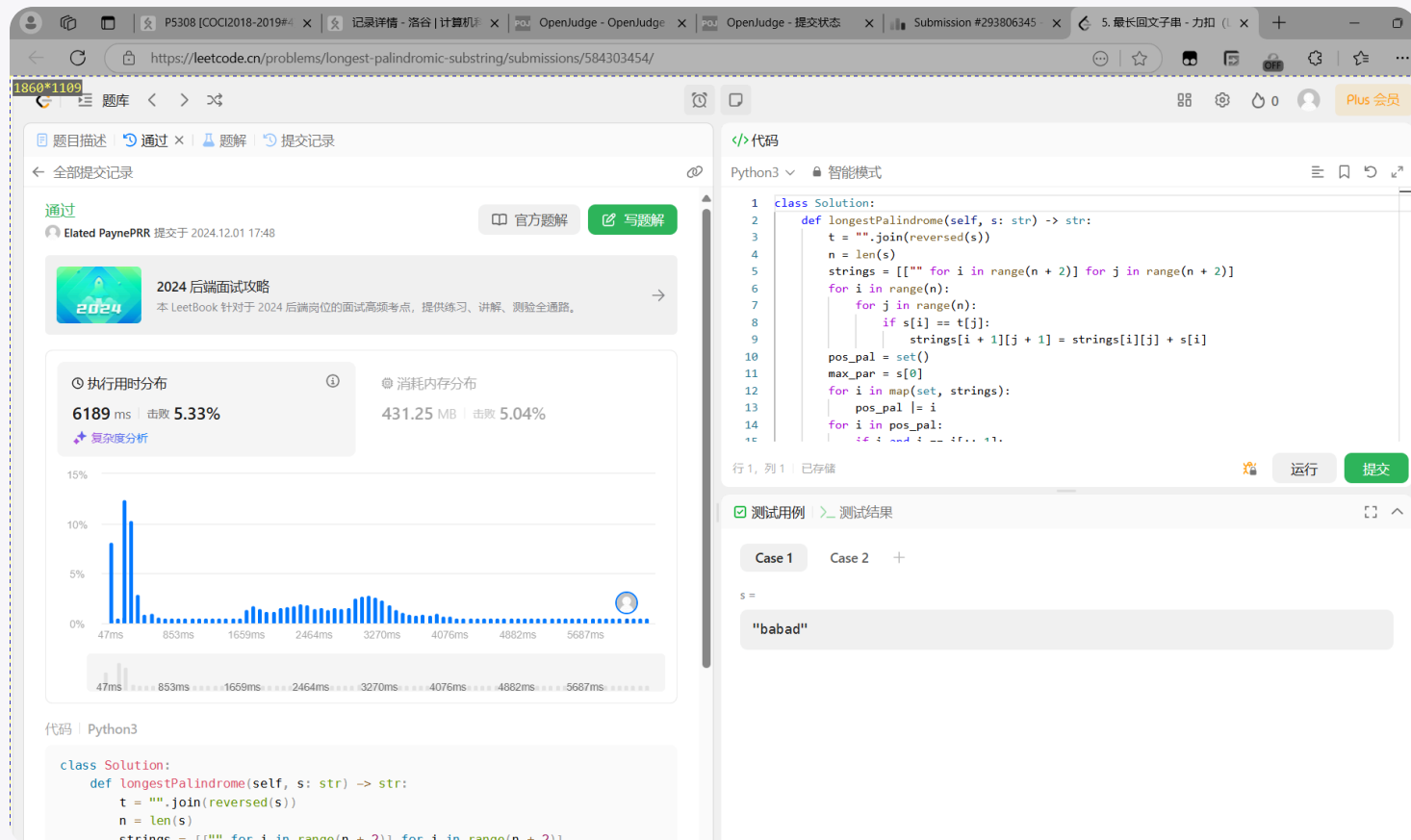
Python

```

17     return max_par
18
19
20 if __name__ == '__main__':
21     sol = Solution()
22     print(sol.longestPalindrome(input()))

```

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：

- 我真的很想看看这个题的测试数据！因为我尝试了一种全新的读取数据思路，即使用 `next()` 对可迭代对象进行操作，这样完全可以丢掉难以操作的下标计数方法，我感觉这玩意实在是太好使了，但我非常不能理解为什么还给我报 **Runtime Error**，差评！

代码：

```

1 from sys import stdin
2 from collections import deque
3 get = tuple(map(int, stdin.read().split()))
4 cnt = 1
5 DIRECTIONS = ((0, 1), (0, -1), (1, 0), (-1, 0))
6
7 def bfs(x, y):
8     if (x, y) == (I, J):
9         return True
10    h = mat[x][y]

```

Python

```

11     queue = deque()
12     queue.append((x, y))
13     while queue:
14         px, py = queue.popleft()
15         for dx, dy in DIRECTIONS:
16             nx, ny = px + dx, py + dy
17             if mat[nx][ny] < h:
18                 if (nx, ny) == (I, J):
19                     return True
20                 queue.append((nx, ny))
21                 mat[nx][ny] = h
22
23 for _ in range(get[0]):
24     yes = False
25     M, N = get[cnt], get[cnt + 1]
26     cnt += 2
27     mat = [[1e9 for i in range(N + 2)]] + [[1e9] + [0 for i in range(N)] +
28     [1e9] for j in range(M)] + [[1e9 for i in range(N + 2)]]
29     for i in range(1, 1 + M):
30         for j in range(1, 1 + N):
31             mat[i][j] = get[cnt + i * M - M + j - 1]
32     cnt += M * N
33     I, J = get[cnt], get[cnt + 1]
34     cnt += 2
35     for i in range(get[cnt]):
36         if bfs(get[cnt + 2 * i + 1], get[cnt + 2 * i + 2]):
37             yes = True
38             break
39     print("Yes" if yes else "No")
40     cnt += 2 * get[cnt] + 1

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from sys import stdin
from collections import deque
get = tuple(map(int, stdin.read().split()))
cnt = 1
DIRECTIONS = ((0, 1), (0, -1), (1, 0), (-1, 0))

def bfs(x, y):
    if (x, y) == (I, J):
        return True
    h = mat[x][y]
    queue = deque()
    queue.append((x, y))
    while queue:
        px, py = queue.popleft()
        for dx, dy in DIRECTIONS:
            nx, ny = px + dx, py + dy
            if mat[nx][ny] < h:
                if (nx, ny) == (I, J):
                    return True
                queue.append((nx, ny))
                mat[nx][ny] = h

for _ in range(get[0]):
    yes = False
    M, N = get[cnt], get[cnt + 1]
    cnt += 2
    mat = [[1e9 for i in range(N + 2)]] + [[1e9] + [0 for i in range(N)]
    for i in range(1, 1 + M):
        for j in range(1, 1 + N):
            mat[i][j] = get[cnt + i * M - M + j - 1]
    cnt += M * N
    I, J = get[cnt], get[cnt + 1]
    cnt += 2
    for i in range(get[cnt]):
        if bfs(get[cnt + 2 * i + 1], get[cnt + 2 * i + 2]):
            yes = True
            break
    print("Yes" if yes else "No")
    cnt += 2 * get[cnt] + 1
```

基本信息

#: 47488417  
题目: 12029  
提交人: 颜鼎盛(24n2400011125)  
内存: 6460kB  
时间: 47ms  
语言: Python3  
提交时间: 2024-11-30 22:24:42

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

## 02802: 小游戏

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

思路:

- 最开始写深搜报超时
- 后来改宽搜报超内存
- 给宽搜用上了堆, 简单剪了个枝, 过了
- 给深搜剪了个枝, 没过
- 无语了

代码:

```
1 from heapq import heappop, heappush
2 DIRECTIONS = ((1, 0), (-1, 0), (0, 1), (0, -1))
3
4 def bfs(x1, y1, x2, y2):
5     min_heap = []
6     min_seg = 1e9
7     min_heap.append((0, x1, y1, {(x1, y1)}, (0, 0)))
8     while min_heap:
9         seg, x, y, visited, last_dir = heappop(min_heap)
10        for dx, dy in DIRECTIONS:
11            nx, ny = x + dx, y + dy
12            if (nx, ny) not in visited:
13                if (nx, ny) == (x2, y2):
14                    min_seg = min(min_seg, seg + (1 if (dx, dy) != last_dir
else 0))
```

Python

```

15         break
16         if board[nx][ny] == " ":
17             if seg + (1 if (dx, dy) != last_dir else 0) < min_seg:
18                 heappush(min_heap, (seg + (1 if (dx, dy) != last_dir
else 0), nx, ny, visited | {(nx, ny)}, (dx, dy)))
19         return min_seg
20
21 for _ in range(1, int(1e9)):
22     w, h = map(int, input().split())
23     if w == 0:
24         break
25     print(f"Board #{_}:")
26     board = [["X" for i in range(w + 4)]]\
27         + [["X"] + [" " for i in range(w + 2)] + ["X"]]\
28         + [["X", " "] + list(input()) + [" ", "X"] for i in range(h)]\
29         + [["X"] + [" " for i in range(w + 2)] + ["X"]]\
30         + [["X" for i in range(w + 4)]]
31     for cnt in range(1, int(1e9)):
32         x1, y1, x2, y2 = map(lambda t: int(t) + 1, input().split())
33         if x1 == 1:
34             break
35         min_seg = bfs(x1, y1, x2, y2)
36         if min_seg == 1e9:
37             print(f"Pair {cnt}: impossible.")
38         else:
39             print(f"Pair {cnt}: {min_seg} segments.")
40     print()

```

## 代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

from heapq import heappop, heappush
DIRECTIONS = ((1, 0), (-1, 0), (0, 1), (0, -1))

def bfs(x1, y1, x2, y2):
    min_heap = []
    min_seg = 1e9
    min_heap.append((0, x1, y1, {(x1, y1)}, (0, 0)))
    while min_heap:
        seg, x, y, visited, last_dir = heappop(min_heap)
        for dx, dy in DIRECTIONS:
            nx, ny = x + dx, y + dy
            if (nx, ny) not in visited:
                if (nx, ny) == (x2, y2):
                    min_seg = min(min_seg, seg + (1 if (dx, dy) != last_dir else 0))
                    break
                if board[ny][nx] == " ":
                    if seg + (1 if (dx, dy) != last_dir else 0) < min_seg:
                        heappush(min_heap, (seg + (1 if (dx, dy) != last_dir else 0), nx, ny, visited | {(nx, ny)}, (dx, dy)))
    return min_seg

for _ in range(1, int(1e9)):
    w, h = map(int, input().split())
    if w == 0:
        break
    print(f"Board #{_}:")
    board = [["X" for i in range(w + 4)]]\
        + [["X"] + [" " for i in range(w + 2)] + ["X"]]\
        + [["X", " "] + list(input()) + [" ", "X"] for i in range(h)]\
        + [["X"] + [" " for i in range(w + 2)] + ["X"]]\
        + [["X" for i in range(w + 4)]]
    for cnt in range(1, int(1e9)):
        x1, y1, x2, y2 = map(lambda t: int(t) + 1, input().split())
        if x1 == 1:
            break
        min_seg = bfs(x1, y1, x2, y2)
        if min_seg == 1e9:
            print(f"Pair {cnt}: impossible.")
        else:
            print(f"Pair {cnt}: {min_seg} segments.")
    print()

```

基本信息

#: 47500106  
 题目: 02802  
 提交人: 颜鼎盛(24n2400011125)  
 内存: 5120kB  
 时间: 151ms  
 语言: Python3  
 提交时间: 2024-12-01 17:16:04



## 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。

第四题感觉是要我命了，当然三五六也相当糟糕

对动态规划的题目掌握不好，对搜索模板的熟练度有待加强。贪心？全方面完蛋！

每次看到有同学说自己计概某个题做了一个下午，我就会想：就这？我都是以天为单位计数的！