

Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Compiled by 邱泽霖 化学与分子工程学院

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:

斐波那契数列，可递归可dp。用时15min

代码:

```
from math import sqrt
import functools
@functools.lru_cache(maxsize=None)
def FibonacciAcc(n):
    if n < 2:
        return 1
    else:
        return FibonacciAcc(n-1) + FibonacciAcc(n-2)
list=[]
for i in range(5001):
    list.append(FibonacciAcc(i))
k=int(input())
print(list[k])
```

代码运行截图 (至少包含有"Accepted")

Python 3293B195ms5.41MB

测试点信息源代码

源代码

```
from math import sqrt
import functools
@functools.lru_cache(maxsize=None)
def FibonacciAcc(n):
    if n < 2:
        return 1
    else:
        return FibonacciAcc(n-1) + FibonacciAcc(n-2)
list=[]
for i in range(5001):
    list.append(FibonacciAcc(i))
k=int(input())
print(list[k])
```

Feishengwuqwq

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 16:32:33

27528: 跳台阶

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

思路:

用数学知识可得答案为2^(n-1), 用时10s

代码:

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

代码运行截图 == (至少包含有"Accepted") ==

#47409243提交状态

查看提交统计提示

状态: Accepted

源代码

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

基本信息

#: 47409243

题目: 27528

提交人: 24n2400011884

内存: 3600kB

时间: 29ms

语言: Python3

提交时间: 2024-11-26 16:36:14

©2002-2022 POJ 京ICP备20010980号-1

English帮助关

474D. Flowers

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


思路:

dp, 直接求和计算速度应该更快, 一开始没看懂modulo1000000007是什么意思, 浪费了一些时间。用时1h

代码：

```
import functools
@functools.lru_cache()
def main():
    t,k=map(int,input().split())
    ans=[i for i in range(k)]
    ans.append(k+1)
    for i in range(t):
        a,b=map(int,input().split())
        for i in range(len(ans), b + 2):
            ans.append((2*ans[-1]- ans[-2] + ans[-k]-ans[-k-1])%1000000007)
        ans=ans[b]-ans[a-1]
        while ans<0:
            ans+=1000000007
        print(ans)
main()
```

代码运行截图 (至少包含有"Accepted")

CODEFORCES
Sponsored by TON

🔔

🇬🇧

🇷🇺

FeishengwuQWQ | [Logout](#)

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		
293454319	Practice: FeishengwuQWQ	474D - 15	Python 3	Accepted	640 ms	5620 KB	2024-11-27 10:08:47	2024-11-27 10:08:47	★	Compare

→ Source

Copy

```
import functools
@functools.lru_cache()
def main():
    t,k=map(int,input().split())
    ans=[i for i in range(k)]
    ans.append(k+1)
    for i in range(t):
        a,b=map(int,input().split())
        for i in range(len(ans), b + 2):
            ans.append((2*ans[-1]- ans[-2] + ans[-k]-ans[-k-1])%1000000007)
        ans=ans[b]-ans[a-1]
        while ans<0:
            ans+=1000000007
        print(ans)
main()
```

[Click](#) to see test details

LeetCode5.最长回文子串

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

思路：

先找到形如aa或aba型的子串，再向两边拓展。用时40min

代码：

```
class Solution(object):
    def longestPalindrome(self, s):
        if len(s)!=1:
            st=[]
```

```

en=[]
for i in range(1,len(s)):
    if s[i]==s[i-1]:
        st.append(i-1)
        en.append(i)
    if s[i]==s[i-2] and i!=1:
        st.append(i-2)
        en.append(i)
ans=0
if len(st)==0:
    return s[0]
for i in range(len(st)):
    stp=st[i]
    enp=en[i]
    lenr=enp-stp+1
    while True:
        stp-=1
        enp+=1
        if 0<=stp<enp<=len(s)-1:
            if s[stp]==s[enp]:
                lenr+=2
            else:
                break
        else:
            break
    if lenr>ans:
        ans=lenr
        ansstr=s[stp+1:enp]
    return ansstr
else:
    return s

```

代码运行截图 (至少包含有"Accepted")

[题库](#)
[<](#)
[>](#)
[🔍](#)

[📖 题目描述](#)
[👤 题解](#)
[📝 提交记录](#)

5. 最长回文子串

已解答

中等 相关标签 相关企业 提示 Aa

给你一个字符串 `s`，找到 `s` 中最长的 [回文子串](#)。

示例 1:

输入: `s = "babad"`
 输出: `"bab"`
 解释: `"aba"` 同样是符合题意的答案。

示例 2:

输入: `s = "cbbd"`
 输出: `"bb"`

提示:

- `1 <= s.length <= 1000`
- `s` 仅由数字和英文字母组成

面试中遇到过这道题? 1/5

👍 7.4K 🗨 5.9K ☆ 🔄 ⓘ

208 人在线

代码

Python 智能模式

```

1 class Solution(object):
2     def longestPalindrome(self, s):
3         if len(s)!=1:
4             st=[]
5             en=[]
6             for i in range(1,len(s)):
7                 if s[i]==s[i-1]:
8                     st.append(i-1)
9                     en.append(i)
10                if s[i]==s[i-2] and i!=1:
11                    st.append(i-2)
                    st.append(i-2)

```

行 4, 列 18 | 已存储 运行 提交

测试用例 测试结果

Case 1 Case 2 +

s =

"babad"

12029: 水淹七军

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

思路：

感觉输入比较麻烦，于是用了许久未用的C++,bfs,一开始反复声明了列表map和maptf导致报错，后看了推荐代码后将列表声明放到main()函数前，就AC了，感谢老师的耐心指导，用时4h

(*´▽`)~♥

代码：

```
#include <iostream>
using namespace std;
#include<queue>
int map[203][203];
bool maptf[203][203];
int main() {
    int dir[4][2] = { {0, 1}, {1, 0}, {0, -1}, {-1, 0} }; int K; cin >> K;
    for (int i = 0; i < K; i++) {
        queue<pair<int, int>>source;///int** map = new int* [203];bool** maptf =
new bool* [203]///;
        ///for (int i = 0; i < 203; ++i) {
            ///map[i] = new int[203];maptf[i] = new bool[203];}
        int M, N; cin >> M >> N;
        for (int j = 0; j <= N + 1; j++) {
            map[0][j] = map[M + 1][j] = 1003; maptf[0][j] = maptf[M + 1][j] =
false;
        }
        for (int j = 0; j <= M + 1; j++) {
            map[j][0] = map[j][N + 1] = 1003; maptf[j][0] = maptf[j][N + 1] =
false;
        }
        for (int k = 1; k <= M; k++) {
            for (int l = 1; l <= N; l++) { cin >> map[k][l]; }
        }
        for (int tx = 0; tx <= M + 1; tx++) {
            for (int ty = 0; ty <= N + 1; ty++) { maptf[tx][ty] = false; }
        }
        int I, J; cin >> I >> J; int P; cin >> P;
        for (int p = 0; p < P; p++) {
            int x, y;
            cin >> x >> y;
            source.push({ x,y });
        }
        while (not source.empty()) {
            pair<int, int> loc = source.front(); int x = loc.first; int y =
loc.second; source.pop();
            maptf[x][y] = true;
            if (x == I and y == J) { cout << "Yes" << endl;; goto finished; }
            for (int dn = 0; dn < 4; dn++) {
                int nx = x + dir[dn][0];
```

```

        int ny = y + dir[dn][1];
        if (map[nx][ny] < map[x][y]) {
            map[nx][ny] = map[x][y];
            source.push({ nx,ny });
        }
    }
}

cout << "No" << endl;
finished:continue;
    ///for (int i = 0; i < 203; ++i) {delete[] map[i];delete[]
maptf[i];}delete[] map;delete[] maptf;
    return 0;
}
}

```

代码运行截图 (至少包含有"Accepted")

#47459411提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```

#include <iostream>
using namespace std;
#include<queue>
int map[203][203];
bool maptf[203][203];
int main() {
    int dir[4][2] = { {0, 1}, {1, 0}, {0, -1}, {-1, 0} }; int K; cin >>
    for (int i = 0; i < K; i++) {
        queue<pair<int, int>>source;///int** map = new int* [203];bool*
        ///for (int i = 0; i < 203; ++i) {
            ///map[i] = new int[203];maptf[i] = new bool[203];}
        int M, N; cin >> M >> N;
        for (int j = 0; j <= N + 1; j++) {
            map[0][j] = map[M + 1][j] = 1003; maptf[0][j] = maptf[M + 1]
        }
        for (int j = 0; j <= M + 1; j++) {
            map[j][0] = map[j][N + 1] = 1003; maptf[j][0] = maptf[j][N
        }
        for (int k = 1; k <= M; k++) {
            for (int l = 1; l <= N; l++) { cin >> map[k][l]; }
        }
        for (int tx = 0; tx <= M + 1; tx++) {
            for (int ty = 0; ty <= N + 1; ty++) { maptf[tx][ty] = false
        }
        int I, J; cin >> I >> J; int P; cin >> P;
    }
}

```

基本信息

#: 47459411
 题目: 12029
 提交人: 24n2400011884
 内存: 420kB
 时间: 18ms
 语言: G++
 提交时间: 2024-11-29 13:00:11

02802: 小游戏

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

思路:

bfs, 每一步向前走时判断与前一步方向一不一样, 不一样的话线段数+1。中途遇到了浅拷贝问题, 但在之前的经验下成功解决, 感觉还是有不少的进步。用时1h

代码:

```

from queue import Queue
import copy
dir=[[1,0],[0,1],[-1,0],[0,-1]]
def check(x,y,m,n):
    if 0<=x<m+2 and 0<=y<n+2:

```

```

        return True
    else:
        return False
def move(x,y,ditu,q,w,h):
    #print(x,y)
    for i in range(4):
        xp=x+dir[i][0]
        yp=y+dir[i][1]
        if check(xp,yp,h,w):
            #print(1)
            if ditu[xp][yp][0]==0:
                if ditu[xp][yp][1] == False:
                    ditu[xp][yp][3] = i
                    if ditu[x][y][3]==i:
                        ditu[xp][yp][1]=True
                        ditu[xp][yp][2]=ditu[x][y][2]
                    else:
                        ditu[xp][yp][1]=True
                        ditu[xp][yp][2]=ditu[x][y][2]+1
                    q.put((xp,yp))
                    #print(xp,yp)
                else:
                    if ditu[x][y][3] == i:
                        if ditu[xp][yp][2] > ditu[x][y][2]:
                            ditu[xp][yp][2]=ditu[x][y][2]
                            q.put((xp,yp))
                            ditu[xp][yp][3]=i
                        else:
                            if ditu[xp][yp][2] > ditu[x][y][2] + 1:
                                ditu[xp][yp][2]=ditu[x][y][2]+1
                                q.put((xp, yp))
                                ditu[xp][yp][3]=i
            return ditu,q
bn=0
while True:
    w,h=map(int,input().split())
    if w==0 and h==0:
        break
    bn+=1
    print("Board #"+str(bn)+":")
    pn=1
    ditu=[0]*(2+w)
    for i in range(h):
        inp=input()
        ditup=[]
        for j in range(w):
            if inp[j]=='X':
                ditup.append(1)
            else:
                ditup.append(0)
        ditu.append([0]+ditup+[0])
    ditu.append([0]*(2+w))
    #print(ditu)
    for i in range(h+2):
        for j in range(w+2):
            ditu[i][j]=[ditu[i][j],False,0,-1]

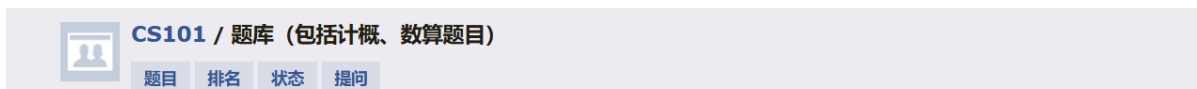
```

```

while True:
    y1,x1,y2,x2=map(int,input().split())
    if x1==x2==y1==y2==0:
        break
    if x1==x2 and y1==y2:
        print("Pair "+str(pn)+": 0 segments.")
        pn+=1
    else:
        ditu1=copy.deepcopy(ditu)
        ditu1[x1][y1][0]=ditu1[x2][y2][0]=0
        q = Queue()
        q.put((x1,y1))
        while not q.empty():
            x,y=q.get()
            ditu1,q=move(x,y,ditu1,q,w,h)
            #print(ditu1)
        if ditu1[x2][y2][2]!=0:
            print("Pair "+str(pn)+": "+str(ditu1[x2][y2][2])+" segments.")
        else:
            print("Pair "+str(pn)+": impossible.")
        pn+=1
print()

```

代码运行截图 (至少包含有"Accepted")



#47461385提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```

from queue import Queue
import copy
dir=[[1,0],[0,1],[-1,0],[0,-1]]
def check(x,y,m,n):
    if 0<=x<=m+2 and 0<=y<=n+2:
        return True
    else:
        return False
def move(x,y,ditu,q,w,h):
    #print(x,y)
    for i in range(4):
        xp=x+dir[i][0]
        yp=y+dir[i][1]
        if check(xp,yp,h,w):
            #print(1)
            if ditu[xp][yp][0]==0:
                if ditu[xp][yp][1] == False:
                    ditu[xp][yp][3] = i
                    if ditu[x][y][3]==i:
                        ditu[xp][yp][1]=True
                        ditu[xp][yp][2]=ditu[x][y][2]
                    else:
                        ditu[xp][yp][1]=True
                        ditu[xp][yp][2]=ditu[x][y][2]+1
                q.put((xp,yp))

```

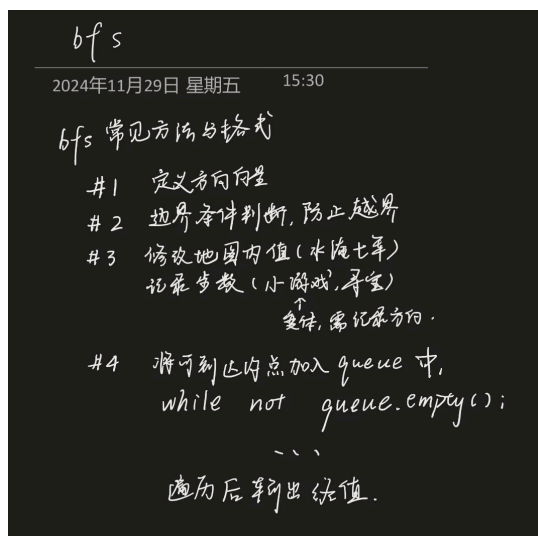
基本信息

#: 47461385
 题目: 02802
 提交人: 24n2400011884
 内存: 6612kB
 时间: 325ms
 语言: Python3
 提交时间: 2024-11-29 15:17:10

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。

这周题目不算太难，但最后两题搜索还是写了比较久，于是自己做了bfs相关的整理



这周还是学到了非常多，如C++要尽量避免反复声明变量，学习了C++与python中队列的使用，了解了浮点型与整型的范围区别，在群里问问题都能获得很有价值的回答，收获很大。