

Assignment #10: dp & bfs

Updated 2 GMT+8 Nov 25, 2024

2024 fall, Complied by 俞天麒 物理学院

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) 如果不能在截止单前提交作业，请写明原因。

1. 题目

LuoguP1255 数楼梯

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

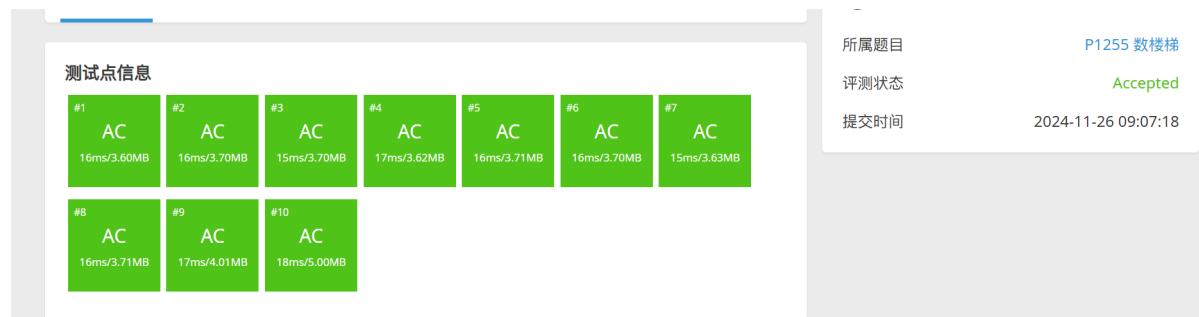
思路:

转移方程 $a[i]=a[i-1]+a[i-2]$, 边界条件 $a[1]=1, a[2]=2$

代码:

```
n=int(input())
a=[0]*n
if n==1:
    print(1)
else:
    a[0],a[1]=1,2
    for i in range(2,n):
        a[i]=a[i-1]+a[i-2]
    print(a[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：

找规律，发现就是 2^{n-1} 。现在证明一下，跳上n级台阶的方法数的转移方程应该是 $a[n]=\sum(a[1:n])+1$ ，边界条件是 $a[1]=1$ ，用数学归纳法可以证明成立。

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))
```

基本信息

#: 47399432

题目: 27528

提交人: 24n2400011425

内存: 3604kB

时间: 30ms

语言: Python3

提交时间: 2024-11-26 09:10:56

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

474D. Flowers

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

思路：

转移方程是 $a[i]=a[i-1]+a[i-2]$ ，表明可以选择通过i-2加上两个W达到，也可以通过i-1加上R达到。还要注意用sum求和很慢，可以设置一个储存答案累积到目前的和的数组，把求和的时间复杂度优化到O(1)

代码：

```
t,k=map(int,input().split())
ans=[0]*(10**5+1)
presum=[0]*(10**5+1)
ans[0]=1
for i in range(1,k):
    ans[i]=1
    presum[i]=presum[i-1]+ans[i]
for i in range(k,10**5+1):
    ans[i]=(ans[i-1]+ans[i-k])%(10**9+7)
    presum[i]=(presum[i-1]+ans[i])%(10**9+7)
for _ in range(t):
    a,b=map(int,input().split())
    #print(ans[a:b+1],presum[a:b+1])
    print((presum[b]-presum[a-1])%(10**9+7))
```

代码运行截图 (至少包含有"Accepted")

The screenshot shows a LeetCode submission details page. The general information table includes:

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged
293047361	Practice: Chirnko	474D - 15	Python 3	Accepted	656 ms	7384 KB	2024-11-24 10:59:04	2024-11-24 10:59:04

The source code is as follows:

```
t,k=map(int,input().split())
ans=[0]*(10**5+1)
presum=[0]*(10**5+1)
ans[0]=1
for i in range(1,k):
    ans[i]=1
    presum[i]=(presum[i-1]+ans[i])
for i in range(k,10**5+1):
    ans[i]=(ans[i-1]+ans[i-k])%(10**9+7)
    presum[i]=(presum[i-1]+ans[i])%(10**9+7)
for i in range(k+1):
    a,b=map(int,input().split())
    #print(ans[a:b+1],presum[a:b+1])
    print((presum[b]-presum[a-1])%(10**9+7))
```

[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：

求助了chatgpt，学习了中心扩展法。即找到一个位置，在左右相同的时候往左右延伸，注意区分奇数偶数长度的情形。

代码：

```
class Solution:
    def longestPalindrome(self, s: str) -> str:
        start,end=0,0
        def expand(i,j,s):
            while i>=0 and j<=len(s)-1 and s[i]==s[j]:
                i-=1
                j+=1
            return i+1,j-1
        for i in range(len(s)-1):
            left,right=expand(i,i,s)
            if right-left>end-start:
                start,end=left,right
            left,right=expand(i,i+1,s)
            if right - left > end - start:
                start, end = left, right
        return (s[start:end+1])
```

代码运行截图 (至少包含有"Accepted")

通过

Practical PascalGNS 提交于 2024.11.26 09:40

执行用时分布
283 ms | 击数 71.97%
消耗内存分布
17.02 MB | 击数 52.79%

```

class Solution:
    def longestPalindrome(self, s: str) -> str:
        start,end=0,0
        def expand(i,j,s):
            while i>=0 and j<len(s)-1 and s[i]==s[j]:
                i-=1
                j+=1
            return i+1,j-1
        for i in range(len(s)-1):
            left,right=expand(i,i,s)
            if right-left>end-start:
                start,end=left,right
        return s[start:end+1]

```

通过
Case 1 Case 2
输入
输出
预期结果

12029: 水淹七军

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

思路：

对于每一个水，遍历能淹的地方，如果那一格的水没有此时的水深，那就改成现在的水。

代码：

```

import sys
origin=sys.stdin.read()
sys.setrecursionlimit(30000000)
data=origin.split()
ind=1
k=int(data[0])
dire=[[0,1],[1,0],[0,-1],[-1,0]]
result=[]
for _ in range(k):
    m,n=map(int,data[ind:ind+2])
    ind+=2
    h=[]
    water=[[-1]*n for i in range(m)]
    for _ in range(m):
        h.append(list(map(int,data[ind:ind+n])))
        ind+=n
    I,J=map(int,data[ind:ind+2])
    ind+=2
    p=int(data[ind])
    ind+=1
    def find(i,j,m,n,hw):
        #print(i,j)
        water[i][j]=hw
        for k in range(4):
            x,y=i+dire[k][0],j+dire[k][1]

```

```

        if x>=0 and x<m and y>=0 and y<n and h[x][y]<=hw and water[x][y]<hw:
            find(x,y,m,n,hw)
for _ in range(p):
    x,y=map(int,data[ind:ind+2])
    ind+=2
    if h[x-1][y-1]>water[x-1][y-1]:
        find(x-1,y-1,m,n,h[x-1][y-1])
#print(water)
result.append("No" if h[I-1][J-1]>=water[I-1][J-1] else "Yes")
sys.stdout.write("\n".join(result)+ "\n")

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```

import sys
origin=sys.stdin.read()
sys.setrecursionlimit(30000000)
data=origin.split()
ind=1
k=int(data[0])
dire=[[0,1],[1,0],[0,-1],[-1,0]]
result=[]
for _ in range(k):
    m,n=map(int,data[ind:ind+2])
    ind+=2
    h=[]
    water=[[-1]*n for i in range(m)]
    for _ in range(m):
        h.append(list(map(int,data[ind:ind+n])))
        ind+=n
    I,J=map(int,data[ind:ind+2])
    ind+=2
    p=int(data[ind])
    ind+=1
    def find(i,j,m,n,hw):
        #print(i,j)
        water[i][j]=hw
        for k in range(4):
            x,y=i+dire[k][0],j+dire[k][1]
            if x>=0 and x<m and y>=0 and y<n and h[x][y]<=hw and water[x][y]<hw:
                find(x,y,m,n,hw)
    for _ in range(p):
        x,y=map(int,data[ind:ind+2])
        ind+=2
        if h[x-1][y-1]>water[x-1][y-1]:
            find(x-1,y-1,m,n,h[x-1][y-1])
#print(water)
result.append("No" if h[I-1][J-1]>=water[I-1][J-1] else "Yes")
sys.stdout.write("\n".join(result)+ "\n")

```

基本信息

#: 47403958
 题目: 12029
 提交人: 24n2400011425
 内存: 18704kB
 时间: 261ms
 语言: Python3
 提交时间: 2024-11-26 12:24:12

02802: 小游戏

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

思路:

真难吧，利用一个东西标记当前行走方向，如果下一个搜索的方向不同，则给turn+1，还要用一个dp数组来储存以简化时间复杂度。超级超级多的细节包括边界，行和列，初始条件什么的要注意，很难。

代码:

```

import sys
sys.setrecursionlimit(3000000)
mark=0
dire=[[0,1],[1,0],[0,-1],[-1,0]]

```

```

while True:
    result=[]
    mark+=1
    w,h=map(int,input().split())
    if w+h==0:
        break
    a=[[0]*(w+2) for i in range(h+2)]
    for i in range(1,h+1):
        s=input()
        for j in range(1,w+1):
            if s[j-1]=="X":
                a[i][j]=1
    while True:
        visit=[[False]*(w+2) for i in range(h+2)]
        x1,y1,x2,y2=map(int,input().split())
        if x1+y1+x2+y2==0:
            break
        a[y1][x1],a[y2][x2]=0,0
        dp=[[float("inf")]*(w+2) for i in range(h+2)]
        def find(i,j,mark,turn):
            #print(i,j,mark,turn)
            if dp[j][i]<=turn:
                return float("inf")
            dp[j][i]=turn
            if i==x2 and j==y2:
                return turn
            mim=float("inf")
            for k in range(4):
                cx,cy=i+dire[k][0],j+dire[k][1]
                if cx>=0 and cx<=w+1 and cy>=0 and cy<=h+1 and visit[cy][cx]==False and a[cy][cx]==0:
                    visit[cy][cx]=True
                    new_turn=turn+1 if k!=mark else turn
                    result=find(cx,cy,k,new_turn)
                    mim=min(mim,result)
                    visit[cy][cx]=False
            return mim
        visit[y1][x1]=True
        result.append(find(x1,y1,-1,0))
        #print(result)
        a[y1][x1],a[y2][x2]=1,1
    print("Board #"+str(mark)+":")
    for i in range(len(result)):
        print("Pair "+str(i+1)+": ",end="")
        if result[i]==float("inf"):
            print("impossible.")
        else:
            print(str(result[i])+" segments.")
    print()

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
import sys
sys.setrecursionlimit(3000000)
mark=0
dire=[[0,1],[1,0],[0,-1],[-1,0]]
while True:
    result=[]
    mark+=1
    w,h=map(int,input().split())
    if w==h==0:
        break
    a=[[0]*(w+2) for i in range(h+2)]
    for i in range(1,h+1):
        s=input()
        for j in range(1,w+1):
            if s[j-1]=="X":
                a[i][j]=1
    while True:
        visit=[[False]*(w+2) for i in range(h+2)]
        x1,y1,x2,y2=map(int,input().split())
        if x1+y1+x2+y2==0:
            break
        a[y1][x1],a[y2][x2]=0,0
        dp=[[float("inf")]* (w+2) for i in range(h+2)]
        def find(i,j,mark,turn):
            #print(i,j,mark,turn)
            if dp[j][i]<=turn:
                return float("inf")
            dp[j][i]=turn
            if i==x2 and j==y2:
                return turn
            mim=float("inf")
            for k in range(4):
                cx,cy=i+dire[k][0],j+dire[k][1]
                if cx>=0 and cx<=w+1 and cy>=0 and cy<=h+1 and visit[cy]
```

基本信息

#: 47403928
题目: 02802
提交人: 24n2400011425
内存: 6836kB
时间: 76ms
语言: Python3
提交时间: 2024-11-26 12:17:10

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。

感觉这次的bfs和dfs都很有难度，虽然模板都知道，但是细节的处理上还是得多花心思。debug很吃时间和耐心，所以这次完成所花的时间超级长。。。