

Assignment #D: 十全十美

Updated 1254 GMT+8 Dec 17, 2024

2024 fall, Compiled by <mark>汤伟杰，信息管理系</mark>

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

02692: 假币问题

brute force, <http://cs101.openjudge.cn/practice/02692>

思路：

以前做过，现在再做只记得当时用到了 `all()` 函数，然后思路就出来了。我记得第一次做的时候没想出来怎么暴力枚举，然后去看了题解。这道题能勾起来那段时间做 oj 上题目的痛苦回忆.....（虽然现在也很痛苦但并不快乐着

代码：

```
letter='ABCDEFGHIJKL'
for _ in range(int(input())):
    s=[]
    for _ in range(3):
        l,r,state=input().split()
        s.append((l,r,state))
    for i in letter:
        if all((i in s[j][0] and s[j][2]=='up' or i in s[j][1] and
s[j][2]=='down' or i not in s[j][0]+s[j][1] and s[j][2]=='even') for j in
range(3)):
            print(f'{i} is the counterfeit coin and it is heavy.')

        elif all((i in s[j][0] and s[j][2]=='down' or i in s[j][1] and
s[j][2]=='up' or i not in s[j][0]+s[j][1] and s[j][2]=='even') for j in
range(3)):
            print(f'{i} is the counterfeit coin and it is light.')
```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

状态: Accepted

源代码

```
letter='ABCDEFGHIJKL'
for _ in range(int(input())):
    s=[]
    for _ in range(3):
        l,r,state=input().split()
        s.append((l,r,state))
    for i in letter:
        if all((i in s[j][0] and s[j][2]=='up' or i in s[j][1] and s[j][2]=='down') for j in range(3)):
            print(f'{i} is the counterfeit coin and it is heavy.')
        elif all((i in s[j][0] and s[j][2]=='down' or i in s[j][1] and s[j][2]=='up') for j in range(3)):
            print(f'{i} is the counterfeit coin and it is light.')
```

基本信息

#: 47792459
题目: 02692
提交人: 24n2400016635
内存: 3592kB
时间: 24ms
语言: Python3
提交时间: 2024-12-17 17:39:25

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

01088: 滑雪

dp, dfs similar, <http://cs101.openjudge.cn/practice/01088>

思路:

dp[i][j]表示以位置(i, j)为起点的最长路径。然后想到这肯定与四周的dp有关,要是比四周的高度大,可以直接拿来加1;但是问题是四周的dp值没法在初始化中设置,怎么办呢?就想到了如果dp值暂时没有,我就继续递归这个位置,直到这个位置的dp值被填充了;那退出条件是啥呢?是最低位置的一个,也就是四周都是墙的那个,所以要把高度加一个保护圈,并用all()函数来快速判断,若真的到最低点了,该处的长度就是1。然后每次更新dp就用全局遍历更新最大值。自己写出来了一道dp题目好高兴(虽然用时很长吧)

代码:

```
import sys
sys.setrecursionlimit(1 << 30)
dx,dy=[0,-1,1,0],[-1,0,0,1]
max_v=1
def dfs(x,y,dp):
    global max_v
    if all(s[x+dx[i]][y+dy[i]]>=s[x][y] for i in range(4)):
        dp[x][y]=1
        return

    for i in range(4):
        nx,ny=x+dx[i],y+dy[i]
        if dp[nx][ny]==-1 and s[nx][ny]<s[x][y]:
            dfs(nx,ny,dp)
        if dp[nx][ny]!=-1 and s[nx][ny]<s[x][y]:
            dp[x][y]=max(dp[x][y],1+dp[nx][ny])
            max_v=max(max_v,dp[x][y])

n,m=map(int,input().split())
s=[];protect=[99999]*(m+2)
```

```

s.append(protect)
for _ in range(n):
    s.append([99999]+list(map(int,input().split()))+[99999])
s.append(protect)
# for i in s: print(*i)
# s=[[int(i) for i in input().split()] for _ in range(n)]
dp=[[-1]*(m+2) for _ in range(n+2)]
for i in range(1,n+1):
    for j in range(1,m+1):
        dfs(i,j,dp)
print(max_v)
# for i in dp: print(*i)

```

代码运行截图 ==（至少包含有"Accepted"）==

OpenJudge 题目ID, 标题, 描述 24n2400016635 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#47802269提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```

import sys
sys.setrecursionlimit(1 << 30)
dx, dy=[0,-1,1,0],[-1,0,0,1]
max_v=1
def dfs(x,y,dp):
    global max_v
    if all(s[x+dx[i]][y+dy[i]]>=s[x][y] for i in range(4)):
        dp[x][y]=1
        return
    for i in range(4):
        nx,ny=x+dx[i],y+dy[i]
        if dp[nx][ny]==-1 and s[nx][ny]<s[x][y]:
            dfs(nx,ny,dp)
        if dp[nx][ny]!=-1 and s[nx][ny]<s[x][y]:
            dp[x][y]=max(dp[x][y],1+dp[nx][ny])
    max_v=max(max_v,dp[x][y])

n,m=map(int,input().split())
s=[[],protect=[99999]*(m+2)]
s.append(protect)

```

基本信息

#: 47802269
 题目: 01088
 提交人: 24n2400016635
 内存: 4368kB
 时间: 67ms
 语言: Python3
 提交时间: 2024-12-18 00:17:50

25572: 螃蟹采蘑菇

bfs, dfs, <http://cs101.openjudge.cn/practice/25572/>

思路:

只记录螃蟹的头并对其 bfs，用 pos 记录是横着走还是竖着走，从而利用两种类型确定 ddx 和 ddy（尾巴），在判断是否到终点的时候要同时看头和尾巴的位置是不是 9，或者 0，或者刚开始的 5，所以用集合把所有可能移动的坐标存起来作为条件判断。有一个坑的存头的坐标时横坐标可能会重新更新。

看了题解发现别人代码好简洁。。

代码:

```

from collections import deque
dx,dy=[0,-1,1,0],[-1,0,0,1]
def bfs(x1,y1,pos,final):
    if pos==1:

```

```

        a,b,ddx,ddy=n,n-1,0,1
    else:
        a,b,ddx,ddy=n-1,n,1,0
    q=deque()
    q.append((x1,y1))
    inq=set()
    inq.add((x1,y1))
    while q:
        for _ in range(len(q)):
            x1,y1=q.popleft()
            for i in range(4):
                nx,ny=x1+dx[i],y1+dy[i]
                if 0<=nx<a and 0<=ny<b and (nx,ny) not in inq and \
                    (s[nx][ny],s[nx+ddx][ny+ddy]) in
((0,0),(5,0),(0,5),(0,9),(9,0),(5,9),(9,5)):
                    if (nx,ny)==final or (nx+ddx,ny+ddy)==final:
                        return 'yes'
                    q.append((nx,ny))
                    inq.add((nx,ny))
    return 'no'

n=int(input())
s=[];x1=-1
for i in range(n):
    l=list(map(int,input().split()))
    if 9 in l:
        j=l.index(9)
        final=(i,j)
    if 5 in l and x1!=-1:
        y1=l.index(5)
        x1=i
        pos=1 if y1+1<n and l[y1+1]==5 else 2
    s.append(l)
if n==1 or n==0:
    print('no')
else:
    print(bfs(x1,y1,type,final))

```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

OpenJudge 题目ID, 标题, 描述 24n2400016635 信箱 账号

CS101 / 题库 (包括计概、数算题目)

题目 排名 状态 提问

#47801031提交状态 查看 提交 统计 提问

状态: Accepted

源代码

from collections import deque
dx, dy=[0,-1,1,0],[-1,0,0,1]
def bfs(x1,y1,type,final):
 if type==1:
 a,b,ddx,ddy=n-1,0,1
 else:
 a,b,ddx,ddy=n-1,n,1,0
 q=deque()
 q.append((x1,y1))
 inq=set()
 inq.add((x1,y1))
 while q:
 for _ in range(len(q)):
 x1,y1=q.popleft()
 for i in range(4):
 nx,ny=x1+dx[i],y1+dy[i]
 if 0<=nx<a and 0<=ny<b and (nx,ny) not in inq and \
 (s[nx][ny],s[nx+ddx][ny+ddy]) in ((0,0),(5,0),(
 if (nx,ny)==final or (nx+ddx,ny+ddy)==final:
 return 'yes'
 q.append((nx,ny))
 inq.add((nx,ny))

基本信息

#: 47801031
题目: 25572
提交人: 24n2400016635
内存: 3752kB
时间: 22ms
语言: Python3
提交时间: 2024-12-17 23:05:34

27373: 最大整数

dp, <http://cs101.openjudge.cn/practice/27373/>

思路:

这题放置了好几天没想出来排序之后怎么 dp, 结果再做的时候突然发现每个数字是价格, 长度是占用的空间大小, 这不就是 01 背包吗! 然后一顿操作, 写了排序和二维 dp, 超时; 再写成一维 dp, 还是超时; 优化预处理, 把数组长度大于 m 的全部去掉, 还是超时。。。没办法了, 我问了 gpt, 用了第四行来替换冒泡排序, 就 ac 了。考场上感觉我很难想到是 01 背包, 而且如果超时了估计心态也崩了吧。。。

第四行就是把数字延长 10 倍再按字典序排序, 使用了 lambda 函数, 很巧妙。

代码:

```
m=int(input())  
n=int(input())  
s=list(map(int,input().split()))  
s = sorted(s, key=lambda x: str(x)*10, reverse=True) #GPT  
  
# f=True  
# for i in range(n-1):  
#     f=False  
#     for j in range(n-i-1):  
#         if str(s[j])+str(s[j+1])<str(s[j+1])+str(s[j]):  
#             s[j],s[j+1]=s[j+1],s[j]  
#             f=True  
#     if f==False:  
#         break  
for i in range(n-1,-1,-1):  
    if len(str(s[i]))<=m:  
        s=s[:i+1]  
        break
```

```

# dp=[['']*(m+1) for _ in range(n)]
# for j in range(1,m+1):
#     if j>=len(str(s[0])):
#         dp[0][j]=str(s[0])
# for i in range(1,n):
#     for j in range(1,m+1):
#         if j-len(str(s[i]))>=0 and dp[i-1][j]:
#             dp[i][j]=str(max(int(dp[i-1][j]),int(dp[i-1][j-
len(str(s[i]))]+str(s[i]))))
#         else:
#             dp[i][j]=dp[i-1][j]
# # for i in dp: print(*i)
# if dp[-1][-1]:
#     print(int(dp[-1][-1]))
# else:
#     print('')

dp=['']*(m+1)
for j in range(m+1):
    if j>=len(str(s[0])):
        dp[j]=str(s[0])
for i in range(1,n):
    for j in range(m,0,-1):
        if j-len(str(s[i]))>=0 and dp[j]:
            dp[j]=str(max(int(dp[j]),int(dp[j-int(len(str(s[i])))]+str(s[i]))))
if dp[-1]:
    print(int(dp[-1]))
else:
    print("")

```

代码运行截图 <mark>（至少包含有"Accepted"）</mark>

OpenJudge
题目ID, 标题, 描述
24n2400016635
信箱
账号

CS101 / 题库 (包括计概、数算题目)
题目
排名
状态
提问

#47847404提交状态
查看
提交
统计
提问

状态: Accepted

源代码

```

m=int(input())
n=int(input())
s=list(map(int,input().split()))
s = sorted(s, key=lambda x: str(x)*10, reverse=True)

# f=True
# for i in range(n-1):
#     f=False
#     for j in range(n-i-1):
#         if str(s[j])+str(s[j+1])<str(s[j+1])+str(s[j]):
#             s[j],s[j+1]=s[j+1],s[j]
#             f=True
#     if f==False:
#         break
for i in range(n-1,-1,-1):
    if len(str(s[i]))<=m:
        s=s[:i+1]
        break
# dp=[['']*(m+1) for _ in range(n)]
# for j in range(1,m+1):
#     if j>=len(str(s[0])):
#         dp[0][j]=str(s[0])

```

基本信息
#: 47847404
题目: 27373
提交人: 24n2400016635
内存: 3740kB
时间: 480ms
语言: Python3
提交时间: 2024-12-19 20:00:32

02811: 熄灯问题

brute force, <http://cs101.openjudge.cn/practice/02811>

思路:

(问的 GPT, 这个游戏从小就没想出来)题目有提示是上一行的灯可以通过操作下一行对应位置来关闭, gpt 说可以对第一行开关灯共 64 种情况枚举, 每一种情况会造成一种第一行的灯的初始开关状态, 接着按题目提示的方式对后续关灯的操作方式是唯一的。自己想不到。

其中在第一行的 64 种 gpt 使用的位运算来转换成 0 和 1 数值串来填充, 很巧。

代码:

```
#GPT
dx,dy=[0,0,-1,1,0],[0,-1,0,0,1]
def press(light,x,y):
    for i in range(5):
        nx,ny=x+dx[i],y+dy[i]
        if 0<=nx<5 and 0<=ny<6:
            light[nx][ny]^=1 #异或, 0^1=1,1^1=0
def doit():
    for first_row in range(64):
        s=[row[:] for row in light] #不能用.copy()->仅用于一维列表
        solution=[[0]*6 for _ in range(5)]
        for j in range(6): #第一行有 64 种开关方式, 逐一遍历
            if (first_row >> j)&1:
                solution[0][j]=1
                press(s,0,j)

        for i in range(1,5):
            for j in range(6):
                if s[i-1][j]==1: #根据上一行开着的灯按下一行的按钮
                    solution[i][j]=1
                    press(s,i,j)

        if all(s[4][j]==0 for j in range(6)): #检查最后一行是不是已经熄灭
            for i in solution:
                print(*i)

light=[[int(i) for i in input().split()] for _ in range(5)]
doit()
```

代码运行截图 <mark> (至少包含有"Accepted") </mark>

#47809566提交状态

查看 提交 统计 提问

状态: Accepted

基本信息

#: 47809566
题目: 02811
提交人: 24n2400016635
内存: 3696kB
时间: 24ms
语言: Python3
提交时间: 2024-12-18 11:16:00

源代码

```
dx,dy=[0,0,-1,1,0],[0,-1,0,0,1]
def press(light,x,y):
    for i in range(5):
        nx,ny=x+dx[i],y+dy[i]
        if 0<=nx<5 and 0<=ny<6:
            light[nx][ny]^=1
def doit():
    for first_row in range(64):
        s=[row[:] for row in light]
        solution=[0]*6 for _ in range(5)
        for j in range(6):
            if (first_row >> j)&1:
                solution[0][j]=1
                press(s,0,j)

        for i in range(1,5):
            for j in range(6):
                if s[i-1][j]==1:
                    solution[i][j]=1
                    press(s,i,j)

        if all(s[4][j]==0 for j in range(6)):
            for i in solution:
                print(*i)

light=[int(i) for i in input().split()] for _ in range(5)
doit()
```

©2002-2022 POJ 京ICP备20010980号-1

English 帮助 关于

08210: 河中跳房子

binary search, greedy, <http://cs101.openjudge.cn/practice/08210/>

思路:

aggressive cow 的题解思路, 一个函数用来二分最长的最短距离, 一个函数用来判断这样的距离是否符合步数, 是模板。

代码:

```
def binary():
    l,r=0,L//(n-m+1)
    while l<=r:
        mid=(l+r)//2
        if can_reach(mid):
            l=mid+1
        else:
            r=mid-1
    return r
```

```
def can_reach(mid):
    cnt=0
    curr=s[0]
    for i in range(1,n+2):
        if s[i]-curr>=mid:
            cnt+=1
            curr=s[i]
    return n+1>=cnt>=n-m+1
```

```
L,n,m=map(int,input().split())
s=[]
for _ in range(n):
    s.append(int(input()))
s=[0]+s+[L]
```



```
print(binary())
```

代码运行截图 <mark>（至少包含有"Accepted"） </mark>

#47801656提交状态

查看提交统计提问

状态: Accepted

源代码

```
def binary():
    l, r = 0, L // (n - m + 1)
    while l <= r:
        mid = (l + r) // 2
        if can_reach(mid):
            l = mid + 1
        else:
            r = mid - 1
    return r

def can_reach(mid):
    cnt = 0
    curr = s[0]
    for i in range(1, n + 2):
        if s[i] - curr >= mid:
            cnt += 1
            curr = s[i]
    return n + 1 >= cnt >= n - m + 1

L, n, m = map(int, input().split())
s = []
for _ in range(n):
    s.append(int(input()))
s = [0] + s + [L]
print(binary())
```

基本信息

#: 47801656

题目: 08210

提交人: 24n2400016635

内存: 6380kB

时间: 169ms

语言: Python3

提交时间: 2024-12-17 23:35:01

2. 学习总结和收获

<mark>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。 </mark>

说实话，现在做 dp 的题目有点害怕，怕自己定义不好数组意义也怕找不到怎么转移，但是一旦想出来了心情就会超级舒畅，比如我做出来了滑雪。在草稿纸上写 dfs 的伪代码有助于分析递归的过程，屡试不爽。只是时间会花费很多，比如滑雪和决战双十一，如果给足够多时间我应该可以做得出来，但是考场上估计太紧张了写不出。

假币问题虽然是枚举但是也好难阿，熄灯问题更抽象，从小时候开始就没搞懂过这个东西。期末不知道会考的怎么样，以前最好成绩是 ac4 但那次都是模板题；期末感觉 ac5 非常非常非常非常有难度（对我而言），对新题自己心里没有多少底，希望自己能超长发挥，尽力吧