

Assignment #E: 算法基础

Updated 2223 GMT+8 Dec 12, 2023

2023 fall, Compiled by 钟明衡 物理学院

说明:

本周作业涉及到枚举、贪心、bfs、矩阵，建议提前开始作业，如果耗时太长，直接找答案看。两个题解，经常更新。所以最好从这个链接下载最新的，<https://github.com/GMyhf/2020fall-cs101>。

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, 学号），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、作业评论有md或者doc。

3) 如果不能在截止前提交作业，请写明原因。

编程环境

操作系统: Windows_NT x64 10.0.19045

Python编程环境: Visual Studio Code 1.76.1

C/C++编程环境: Visual Studio Code 1.76.1

1. 题目

如果耗时太长，直接看解题思路，或者源码

02692: 假币问题

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

思路:

由于数据量不大，可以采用枚举

首先，如果even，则两边的所有硬币都是真币，之后只需要枚举那些不一定是真币的硬币。

把可能轻、重的硬币放到同一边，然后分别假定每个不确定的硬币是轻或者重，如果完全符合条件就直接输出，因为解唯一。

代码

```
1 n = int(input())
2 for _ in range(n):
3     left = []
4     right = []
5     dic = {'A': False, 'B': False, 'C': False, 'D': False,
6            'E': False, 'F': False, 'G': False, 'H': False,
7            'I': False, 'J': False, 'K': False, 'L': False}
8     for i in range(3):
9         s = input().split(' ')
10        if s[2] == 'even':
11            for a in s[0]:
12                dic[a] = True
13            for a in s[1]:
14                dic[a] = True
15        else:
16            if s[2] == 'up':
17                left.append(s[0])
18                right.append(s[1])
19            else:
20                left.append(s[1])
21                right.append(s[0])
22        for a in dic.keys():
23            if not dic[a]:
24                b = True
25                for i in left:
26                    if a not in i:
27                        b = False
28                        break
29                if b:
30                    print(a+' is the counterfeit coin and it is heavy.')
31                    break
32                b = True
33                for i in right:
34                    if a not in i:
35                        b = False
36                        break
37                if b:
38                    print(a+' is the counterfeit coin and it is light.')
39                    break
40
```

代码运行截图

状态: Accepted

源代码

```
n = int(input())
for _ in range(n):
    left = []
    right = []
    dic = {'A': False, 'B': False, 'C': False, 'D': False,
           'E': False, 'F': False, 'G': False, 'H': False,
           'I': False, 'J': False, 'K': False, 'L': False}
    for i in range(3):
        s = input().split(' ')
        if s[2] == 'even':
            for a in s[0]:
                dic[a] = True
            for a in s[1]:
                dic[a] = True
        else:
            if s[2] == 'up':
                left.append(s[0])
                right.append(s[1])
            else:
                left.append(s[1])
                right.append(s[0])
    for a in dic.keys():
        if not dic[a]:
            b = True
            for i in left:
                if a not in i:
                    b = False
                    break
            if b:
                print(a+' is the counterfeit coin and it is heavy.')
                break
            b = True
            for i in right:
                if a not in i:
                    b = False
                    break
            if b:
                print(a+' is the counterfeit coin and it is light.')
                break
```

基本信息

#: 41594964

题目: 02692

提交人: 23n2300011505(12号娱乐选手)

内存: 3604kB

时间: 33ms

语言: Python3

提交时间: 2023-10-11 22:11:31

18164: 剪绳子

greedy/huffman, <http://cs101.openjudge.cn/practice/18164/>

思路:

可以反过来看这个问题, 把绳子合起来

由于合成绳子的次数是固定的, 合成以后的绳长会加到答案里面, 所以每次都取目前最短的两端绳子合成即可

用二分把新的绳长插回原来的列表中

代码

```
1 from bisect import bisect
2
3 n = int(input())
4 l = sorted(list(map(int, input().split())))
5 ans = 0
6 for i in range(n-1):
7     new = l[2*i]+l[2*i+1]
8     ans += new
9     l.insert(bisect(l, new), new)
10 print(ans)
11
```

代码运行截图

#43088416提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
from bisect import bisect

n = int(input())
l = sorted(list(map(int, input().split())))
ans = 0
for i in range(n-1):
    new = l[2*i]+l[2*i+1]
    ans += new
    l.insert(bisect(l, new), new)
print(ans)
```

基本信息

#: 43088416

题目: 18164

提交人: 23n2300011505(12号娱乐选手)

内存: 5588kB

时间: 54ms

语言: Python3

提交时间: 2023-12-12 15:46:45

01328: Radar Installation

greedy, <http://cs101.openjudge.cn/practice/01328/>

思路:

可以转化为x轴上的覆盖问题, 要使得每个区间都至少有一个雷达

按照区间的右端从小到大排列, 然后依次放置雷达, 每放置一次就把被覆盖了的区间排除掉即可

代码

```
1 from math import sqrt
2
3 step = 0
4 while True:
5     step += 1
6     n, d = map(int, input().split())
7     if n == d == 0:
```

```

8         break
9     l = [(0, 0)]*n
10    used = [False]*n
11    b = False
12    for i in range(n):
13        x, y = map(int, input().split())
14        if y > d:
15            b = True
16        else:
17            l[i] = (x-sqrt(d**2-y**2), x+sqrt(d**2-y**2))
18    s = input()
19    if b:
20        print('Case %d: -1' % step)
21        continue
22    ans = 0
23    l.sort(key=lambda x: x[1])
24    for i in range(n):
25        if not used[i]:
26            ans += 1
27            used[i] = True
28            for j in range(i+1, n):
29                if l[j][0] <= l[i][1] and not used[j]:
30                    used[j] = True
31    print('Case %d: %d' % (step, ans))
32

```

代码运行截图

状态: Accepted

源代码

```
from math import sqrt

step = 0
while True:
    step += 1
    n, d = map(int, input().split())
    if n == d == 0:
        break
    l = [(0, 0)] * n
    used = [False] * n
    b = False
    for i in range(n):
        x, y = map(int, input().split())
        if y > d:
            b = True
        else:
            l[i] = (x - sqrt(d**2 - y**2), x + sqrt(d**2 - y**2))
    s = input()
    if b:
        print('Case %d: -1' % step)
        continue
    ans = 0
    l.sort(key=lambda x: x[1])
    for i in range(n):
        if not used[i]:
            ans += 1
            used[i] = True
            for j in range(i+1, n):
                if l[j][0] <= l[i][1] and not used[j]:
                    used[j] = True
    print('Case %d: %d' % (step, ans))
```

基本信息

#: 43097364

题目: 01328

提交人: 23n2300011505(12号娱乐选手)

内存: 3688kB

时间: 92ms

语言: Python3

提交时间: 2023-12-12 22:22:12

19930: 寻宝

bfs, <http://cs101.openjudge.cn/practice/19930>

思路:

标准的bfs, 把下一步能走的存进去, step+1然后继续走

代码

```
1 from sys import exit
2
3 m, n = map(int, input().split())
4 M = [[2] * (n+2)]
5 for i in range(m):
6     M.append([2] + list(map(int, input().split())) + [2])
7 M.append([2] * (n+2))
8 if M[1][1] == 1:
9     print(0)
10    exit()
11 step, next = 0, 0
12 x = [1]
13 y = [1]
```

```

14 used = [[False]*(n+2) for i in range(m+2)]
15 used[1][1] = True
16 dx = [1, 0, -1, 0]
17 dy = [0, 1, 0, -1]
18 while next != len(x):
19     step += 1
20     start = next
21     next = len(x)
22     for i in range(start, next):
23         for j in range(4):
24             newx = x[i]+dx[j]
25             newy = y[i]+dy[j]
26             if M[newy][newx] == 1:
27                 print(step)
28                 exit()
29             elif not used[newy][newx] and M[newy][newx] == 0:
30                 x.append(newx)
31                 y.append(newy)
32                 used[newy][newx] = True
33 print('NO')
34

```

代码运行截图

#43089754提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

from sys import exit

m, n = map(int, input().split())
M = [[2]*(n+2)]
for i in range(m):
    M.append([2]+list(map(int, input().split()))+[2])
M.append([2]*(n+2))
if M[1][1] == 1:
    print(0)
    exit()
step, next = 0, 0
x = [1]
y = [1]
used = [[False]*(n+2) for i in range(m+2)]
used[1][1] = True
dx = [1, 0, -1, 0]
dy = [0, 1, 0, -1]
while next != len(x):
    step += 1
    start = next
    next = len(x)
    for i in range(start, next):
        for j in range(4):
            newx = x[i]+dx[j]
            newy = y[i]+dy[j]
            if M[newy][newx] == 1:
                print(step)
                exit()
            elif not used[newy][newx] and M[newy][newx] == 0:
                x.append(newx)
                y.append(newy)
                used[newy][newx] = True

print('NO')

```

基本信息

#: 43089754

题目: 19930

提交人: 23n2300011505(12号娱乐选手)

内存: 3684kB

时间: 25ms

语言: Python3

提交时间: 2023-12-12 16:39:26

1163B2. Cat Party (Hard Edition)

<https://codeforces.com/contest/1163/problem/B2>

好题。通过维护双层（三层？）数据结构可以 $O(n)$ 。

确实好题，而且感觉难度适合作业没有复杂的东西。多维护了几个数就做到 $O(n)$ 了。

思路：

满足要求的有以下四种情况：

1. 只出现了一个数
2. 每个数都只出现一次
3. 有一个数只出现了一次，其他数出现次数相同
4. 有一个数出现次数最多，其他数出现次数相同，都比最多次数少一次

每次只要判断新加入一个数以后是否满足上述四种情况的其中一种即可。

具体方法是，用一个数组记录每个数出现的次数，用另一个数组记录每一个出现次数上的数的数量。后面的这个数组，可以通过每次加入一个数，就从这个数出现次数的前一个的那个位置挪一给现在它出现的位置，这样第零位上自然会保存出现的不同数字个数，便于检查是否所有数都处于（最大次数/次数为一）或者（最大次数/最大次数减一）的位置上。

代码

```
1  n = int(input())
2  l = list(map(int, input().split()))
3  count = [0]*100001
4  num = [0]*100001
5  ans = 0
6  maxn = 0
7  for i in range(n):
8      count[num[l[i]]] -= 1
9      num[l[i]] += 1
10     count[num[l[i]]] += 1
11     maxn = max(maxn, num[l[i]])
12     if count[0] == -1 or maxn == 1 or (count[maxn] == 1 and count[maxn-
13         1]+count[maxn]+count[0] == 0) or (count[1] == 1 and
14         count[maxn]+count[1]+count[0] == 0):
15         ans = i
16 print(ans+1)
```

代码运行截图

General									
#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
236895298	Practice: MinghengZhong	1163B2 - 25	Python 3	Accepted	140 ms	12828 KB	2023-12-12 15:18:46	2023-12-12 15:18:46	★ Compare

→ [Source](#)
Copy

```

n = int(input())
l = list(map(int, input().split()))
count = [0]*100001
num = [0]*100001
ans = 0
maxn = 0
for i in range(n):
    count[num[l[i]]] -= 1
    num[l[i]] -= 1
    count[num[l[i]]] += 1
    maxn = max(maxn, num[l[i]])
    if count[0] == -1 or maxn == 1 or (count[maxn] == 1 and count[maxn-1]+count[maxn]+count[0] == 0) or (count[1] == 1 and count[maxn]+count[1]+count[0] == 0):
        ans = i
print(ans+1)

```

02811: 熄灯问题

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

思路：

理论上可以列方程把答案矩阵解出来，但是太难了

所以枚举答案矩阵的第一列（选择列而不是行是因为行数小于列数），只有64种情况，之后就可以把每一列都按要求唯一地生成出来，再判断最后一列是否符合要求即可

代码

```

1  M = [[0]*8]
2  for _ in range(5):
3      M.append([0]+list(map(int, input().split()))+[0])
4  M.append([0]*8)
5  ans = [[0]*8 for i in range(7)]
6  for k in range(64):
7      for i in range(1, 6):
8          ans[i][1] = k % 2
9          k //= 2
10     for i in range(2, 7):
11         for j in range(1, 6):
12             ans[j][i] = (ans[j][i-1]+ans[j-1][i-1]+ans[j+1]
13                          [i-1]+ans[j][i-2]+M[j][i-1]) % 2
14     b = True
15     for i in range(1, 6):
16         if M[i][6] != (ans[i][6]+ans[i][5]+ans[i-1][6]+ans[i+1][6]) % 2:
17             b = False
18             break
19     if b:
20         for i in range(1, 6):
21             print(' '.join(map(str, ans[i][1:-1])))
22         break
23

```

代码运行截图

#43096663提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
M = [[0]*8]
for _ in range(5):
    M.append([0]+list(map(int, input().split()))+[0])
M.append([0]*8)
ans = [[0]*8 for i in range(7)]
for k in range(64):
    for i in range(1, 6):
        ans[i][1] = k % 2
        k //= 2
    for i in range(2, 7):
        for j in range(1, 6):
            ans[j][i] = (ans[j][i-1]+ans[j-1][i-1]+ans[j+1][i-1]+ans[j][i-2]+M[j][i-1]) % 2
    b = True
    for i in range(1, 6):
        if M[i][6] != (ans[i][6]+ans[i][5]+ans[i-1][6]+ans[i+1][6]) % 2:
            b = False
            break
    if b:
        for i in range(1, 6):
            print(' '.join(map(str, ans[i][1:-1])))
            break
```

基本信息

#: 43096663

题目: 02811

提交人: 23n2300011505(12号娱乐选手)

内存: 3684kB

时间: 22ms

语言: Python3

提交时间: 2023-12-12 21:39:34

02802: 小游戏

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

思路:

采用一个特殊的bfs, 优先搜索线段数少的路径, 其他步骤和传统bfs一致

注意每组数据最后输出空行

代码

```
1 t, n, m, ans = 0, 0, 0, 0
2 dx = [1, 0, -1, 0]
3 dy = [0, 1, 0, -1]
4 used = []
5 M = []
6
7
8 def check(x, y, a, b):
9     global used
10    if x == a and y == b:
11        return True
12    return M[y+1][x+1] == ' ' and not used[y][x]
13
14
```

```

15 def bfs(x, y, a, b):
16     global used, ans
17     used = [[False]*(n+2) for i in range(m+2)]
18     lx = [x]
19     ly = [y]
20     start = 0
21     used[y][x] = True
22     seg = 0
23     while start != len(lx):
24         seg += 1
25         end = len(lx)
26         for i in range(start, end):
27             for k in range(4):
28                 newx = lx[i]+dx[k]
29                 newy = ly[i]+dy[k]
30                 while check(newx, newy, a, b):
31                     if newx == a and newy == b:
32                         ans = seg
33                         return
34                     lx.append(newx)
35                     ly.append(newy)
36                     used[newy][newx] = True
37                     newx = newx+dx[k]
38                     newy = newy+dy[k]
39             start = end
40     return
41
42
43 while True:
44     n, m = map(int, input().split())
45     if n == m == 0:
46         break
47     t += 1
48     M = ['X'*(n+4), 'X'+ ' '*(n+2)+'X']
49     for _ in range(m):
50         M.append('X '+input()+' X')
51     M += ['X'+ ' '*(n+2)+'X', 'X'*(n+4)]
52     print('Board #d:' % t)
53     tt = 0
54     while True:
55         x, y, a, b = map(int, input().split())
56         if x == y == a == b == 0:
57             break
58         tt += 1
59         ans = -1
60         bfs(x, y, a, b)
61         if ans == -1:
62             print('Pair %d: impossible.' % tt)
63         else:
64             print('Pair %d: %d segments.' % (tt, ans))
65     print('')
66

```

代码运行截图

#43096419提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```
t, n, m, ans = 0, 0, 0, 0
dx = [1, 0, -1, 0]
dy = [0, 1, 0, -1]
used = []
M = []

def check(x, y, a, b):
    global used
    if x == a and y == b:
        return True
    return M[y+1][x+1] == ' ' and not used[y][x]

def bfs(x, y, a, b):
    global used, ans
    used = [[False]*(n+2) for i in range(m+2)]
    lx = [x]
    ly = [y]
    start = 0
    used[y][x] = True
    seg = 0
    while start != len(lx):
        seg += 1
        end = len(lx)
        for i in range(start, end):
            for k in range(4):
                newx = lx[i]+dx[k]
                newy = ly[i]+dy[k]
                while check(newx, newy, a, b):
                    if newx == a and newy == b:
                        ans = seg
                        return
                    lx.append(newx)
                    ly.append(newy)
                    used[newy][newx] = True
                    newx = newx+dx[k]
                    newy = newy+dy[k]
            start = end
    return

while True:
    n, m = map(int, input().split())
    if n == m == 0:
        break
    t += 1
    M = ['X'*(n+4), 'X'+' '*(n+2)+'X']
    for _ in range(m):
        M.append('X'+input()+ 'X')
    M += ['X'+' '*(n+2)+'X', 'X'*(n+4)]
    print('Board #d:' % t)
    tt = 0
    while True:
        x, y, a, b = map(int, input().split())
        if x == y == a == b == 0:
            break
        tt += 1
        ans = -1
        bfs(x, y, a, b)
        if ans == -1:
            print('Pair %d: impossible.' % tt)
        else:
            print('Pair %d: %d segments.' % (tt, ans))
    print('')
```

基本信息

#: 43096419

题目: 02802

提交人: 23n2300011505(12号娱乐选手)

内存: 3848kB

时间: 38ms

语言: Python3

提交时间: 2023-12-12 21:26:27

2. 学习总结和收获

收获最大的是最后一题，用了一个非传统的bfs，不是按照步数优先，而是按照路径线段数优先，现在感觉对这个算法有了更深入的理解。

有些题目需要用到优化后的枚举，一般可以这样做出来的题目，也可以用纯数学，但是想出来方法要好久，用枚举就完全够用。