

# Assignment #E: 算法基础

Updated 1419 GMT+8 Dec 12, 2023

2023 fall, Compiled by 同学的姓名、院系

## 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

(请改为同学的操作系统、编程环境等)

操作系统: macOS Ventura 13.4.1 (c)

Python编程环境: Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境: Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

## 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:

遍历

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 for j in range(int(input())):
5     L = [[], [], []]
```

```

6     flag = 0
7     for i in range(3):
8         L[i] = input().split()
9     for f in 'ABCDEFGHijkl':
10        if all((f in i[0] and i[2]=='up') or (f in i[1] and i[2]=='down')
11              or (f not in i[0] + i[1] and i[2]=='even') for i in L):
12            flag = 'heavy'
13            break
14        if all((f in i[0] and i[2]=='down') or (f in i[1] and i[2]=='up')
15              or (f not in i[0]+i[1] and i[2]=='even') for i in L):
16            flag = 'light'
17            break
18    print(f + " is the counterfeit coin and it is "+flag+ ".")

```

代码运行截图

状态: Accepted

源代码

```

'''
刘思瑞 2100017810
'''
for j in range(int(input())):
    L = [[], [], []]
    flag = 0
    for i in range(3):
        L[i] = input().split()
    for f in 'ABCDEFGHijkl':
        if all((f in i[0] and i[2]=='up') or (f in i[1] and i[2]=='down')
              or (f not in i[0] + i[1] and i[2]=='even') for i in L):
            flag = 'heavy'
            break
        if all((f in i[0] and i[2]=='down') or (f in i[1] and i[2]=='up')
              or (f not in i[0]+i[1] and i[2]=='even') for i in L):
            flag = 'light'
            break
    print(f + " is the counterfeit coin and it is "+flag+ ".")

```

## 18164: 剪绳子

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

思路:

贪心, 但是时间好紧

## 代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  import bisect
5  n = int(input())
6  L = list(map(int, input().split()))
7  L.sort()
8  sum = 0
9  for i in range(n-1):
10     sum += L[0] + L[1]
11     bisect.insort(L, L[0]+L[1])
12     L = L[2:]
13 print(sum)
```

## 代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
import bisect
n = int(input())
L = list(map(int, input().split()))
L.sort()
sum = 0
for i in range(n-1):
    sum += L[0] + L[1]
    bisect.insort(L, L[0]+L[1])
    L = L[2:]
print(sum)
```

©2002-2022 POJ 京ICP备20010980号-1

## 01328: Radar Installation

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

思路:

按岛屿 x 坐标排序, 贪心遍历岛屿, 每次选择最优的雷达安装位置以最小化雷达数量, 同时根据当前岛屿更新雷达的覆盖范围

## 代码

```
1  import math
2
3  def minimal_radar_installations(n, d, islands):
4      islands.sort(key=lambda x: x[0])
5      radar_count = 0
6      current_position = -math.inf
```

```

7     for island in islands:
8         x, y = island
9         if y > d:
10            return -1
11        if x - y > current_position:
12            current_position = x + y
13            radar_count += 1
14        elif x + y >= current_position:
15            current_position = x + y
16    return radar_count
17
18    case_number = 0
19    while True:
20        n, d = map(int, input().split())
21        if n == 0 and d == 0:
22            break
23
24        case_number += 1
25        islands = [list(map(int, input().split())) for _ in range(n)]
26        result = minimal_radar_installations(n, d, islands)
27
28        print(f"Case {case_number}: {result}")
29        input()

```

代码运行截图

状态: **Accepted**

源代码

```

import math

def minimal_radar_installations(n, d, islands):
    islands.sort(key=lambda x: x[0])
    radar_count = 0
    current_position = -math.inf
    for island in islands:
        x, y = island
        if y > d:
            return -1
        if x - y > current_position:
            current_position = x + y
            radar_count += 1
        elif x + y >= current_position:
            current_position = x + y
    return radar_count

case_number = 0
rr=0
while True:

```

# 19930: 寻宝

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

思路:

bfs

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  def bfs(q, head, tail, steps):
5      for k in range(head, tail):
6          x, y = q[head]
7          head += 1
8          if (g[x][y] == 1):
9              print(steps)
10             return False, False, False, False
11         for z in range(4):
12             newx = x + step[z][0]
13             newy = y + step[z][1]
14             if (check(newx, newy)):
15                 vis[newx][newy] = 1
16                 q.append((newx, newy))
17                 tail += 1
18         return q, head, tail, steps+1
19  def check(x, y):
20      if (x < 0 or y < 0 or x >= m or y >= n):
21          return False
22      if (vis[x][y] or g[x][y] == 2):
23          return False
24      return True
25
26  q = []
27  step = [[0, 1], [1, 0], [-1, 0], [0, -1]]
28  vis = [[0] * 52 for _ in range(52)]
29  g = []
30  m, n = map(int, input().split())
31  for i in range(m):
32      g.append([int(x) for x in input().split()])
33  q.append((0, 0))
34  head = 0
35  tail = 1
36  steps = 0
37  while True:
38      q, head, tail, steps = bfs(q, head, tail, steps)
39      if not steps:
40          break
41      if head >= tail:
42          print('NO')
43          break
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def bfs(q, head, tail, steps):
    for k in range(head, tail):
        x, y = q[head]
        head += 1
        if (g[x][y] == 1):
            print(steps)
            return False, False, False, False
        for z in range(4):
            newx = x + step[z][0]
            newy = y + step[z][1]
            if (check(newx, newy)):
                vis[newx][newy] = 1
                q.append((newx, newy))
```

## 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：

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  n=int(input())
5  l=list(map(int,input().split()))
6  a,b=[0]*(10**6), [0]*(10**6)
7  ans=1
8  j=1
9  for i in range(0,n):
10     a[l[i]]+=1
11     b[a[l[i]]]+=1
12     if a[l[i]]*b[a[l[i]]]==j and j!=n:
13         ans=j+1
14     elif a[l[i]]*b[a[l[i]]]==j-1:
15         ans=j
```

```

16     j+=1
17     print(ans)

```

代码运行截图 不知道为什么这个一直在队列里面没有跑

Sponsored by TON

By meinvader, contest: Codeforces Round 558 (Div. 2), problem: (B2) Cat Party (Hard Edition), In queue, #, [Copy](#)

```

'''
刘思瑞 2100017810
'''
n=int(input())
l=list(map(int,input().split()))
a,b=[0]*(10**6),[0]*(10**6)
ans=1
j=1
for i in range(0,n):
    a[l[i]]+=1
    b[a[l[i]]]+=1
    if a[l[i]]*b[a[l[i]]]==j and j!=n:
        ans=j+1
    elif a[l[i]]*b[a[l[i]]]==j-1:
        ans=j
    j+=1
print(ans)

```

## 02811: 熄灯问题

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

思路:

遍历第一层

代码

```

1  '''
2  刘思瑞 2100017810
3  '''
4  import copy
5  li = []
6  def erupt(l,i):
7      global li
8      if i == 6:
9          li.append(l)
10         return
11         for j in [0,1]:
12             l.append(j)
13             erupt(l,i+1)
14             l.pop()
15         return
16  x = [[False]*8]
17  Y = [[False]*8]

```

```

18 for _ in range(5):
19     x.append([False] + [bool(x) for x in input().split()] + [False])
20     y.append([[False]*8])
21 x.append([[False]*8])
22 y.append([[False]*8])
23 for l1i in l1:
24     A = copy.deepcopy(X)
25     B = copy.deepcopy(Y)
26     for i in range(1, 7):
27         if B[1][i]:
28             A[1][i] = not A[1][i]
29             A[1][i-1] = not(A[1][i-1])
30             A[1][i+1] = not(A[1][i+1])
31             A[2][i] = not(A[2][i])
32     for i in range(2, 6):
33         for j in range(1, 7):
34             if A[i-1][j]:
35                 B[i][j] = True
36                 A[i][j] = not A[i][j]
37                 A[i-1][j] = not A[i-1][j]
38                 A[i+1][j] = not A[i+1][j]
39                 A[i][j-1] = not A[i][j-1]
40                 A[i][j+1] = not A[i][j+1]
41     if all((not A[5][i] for i in range(1,7))):
42         for i in range(1, 6):
43             print(" ".join(repr(y) for y in [B[i][1],B[i][2],B[i][3],B[i]
[4],B[i][5],B[i][6] ]))

```

代码运行截图

状态: Accepted

基本

源代码

```

'''
刘思瑞 2100017810
'''
import copy
li = []
def erupt(l,i):
    global li
    if i == 6:
        li.append(l)
        return
    for j in [0,1]:
        l.append(j)
        erupt(l,i+1)
        l.pop()
    return
v = [[0,0,0,0,0,0,0,0]]

```

提

提



## 02802: 小游戏

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

思路:

实在想不明白自己的bug在哪里, 先交个标答我再慢慢想 ( ( (

代码

```
1  import sys
2  sys.setrecursionlimit(1000000)
3  d=[(0,-1),(0,1),(-1,0),(1,0)]
4  H,L,ha,la,hb,lb,MIN=0,0,0,0,0,0,0
5  b=0
6  def dfs(h,l,dire,step):
7      global H,L,hb,lb,MIN,b
8      if h==hb and l==lb:
9          if step<MIN:
10             MIN=step
11             return
12         if step>=MIN:
13             return
14         for i in d:
15             hh,ll=h+i[0],l+i[1]
16             if hh>=0 and hh<=H+1 and ll>=0 and ll<=L+1 and b[hh][ll]==' ':
17                 b[hh][ll]='X'
18                 if dire!=i:
19                     dfs(hh,ll,i,step+1)
20             else:
21                 dfs(hh,ll,i,step)
22             b[hh][ll]=' '
23
24  k1=0
25  while True:
26      k1+=1
27      L,H=map(int,input().split())
28      if L==0:
29          break
30      print("Board #{}:".format(k1))
31      b=[[' ']*(L+2)]
32      for _ in range(H):
33          b.append([' ']+list(input())+[' '])
34      b.append([' ']*(L+2))
35      k2=0
36      while True:
37          k2+=1
38          la,ha,lb,hb=map(int,input().split())
39          MIN=float('inf')
40          if la==0:
41              break
42          b[hb][lb]=' '
43          dfs(ha,la,(0,0),0)
44          b[hb][lb]='X'
```

```

45         if MIN==float('inf'):
46             print("Pair {}: impossible.".format(k2))
47         else:
48             print("Pair {}: {} segments.".format(k2,MIN))
49     print()

```

代码运行截图

状态: Accepted

源代码

```

import sys
sys.setrecursionlimit(1000000)
d=[(0,-1),(0,1),(-1,0),(1,0)]
H,L,ha,la,hb,lb,MIN=0,0,0,0,0,0,0
b=0
def dfs(h,l,dire,step):
    global H,L,hb,lb,MIN,b
    if h==hb and l==lb:
        if step<MIN:
            MIN=step
        return
    if step>=MIN:
        return
    for i in d:
        hh,ll=h+i[0],l+i[1]
        if hh>=0 and hh<=H+1 and ll>=0 and ll<=L+1 and b[hh][ll]!='X':
            b[hh][ll]='X'
            if dire!=i:
                dfs(hh,ll,i,step+1)

```

基本信息

- #: 43237048
- 题目: 02802
- 提交人: 23n2100017810
- 内存: 6280kB
- 时间: 45ms
- 语言: Python3
- 提交时间: 2023-12-19 23:33:06

## 2. 学习总结和收获

主要复习了搜索算法和递归算法，还是有思路但是找不到bug的问题