

# Assignment #F: All-Killed 满分

---

Updated 2217 GMT+8 May 21, 2024

2024 spring, Compiled by 钟明衡 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

---

### 22485: 升空的焰火，从侧面看

<http://cs101.openjudge.cn/practice/22485/>

思路:

构建好树后，按照任意顺序遍历树都可以，只要保证右子树比左子树后遍历到即可，为了方便我就用了dfs

记录当前的层数，每次搜索到一个节点，就更新当前层上的结果，这样就可以保证每层越右边的越后搜索到，从而满足要求

代码

```
1 l, r = {}, {}
2 ans = []
3 for i in range(int(input())):
4     l[i+1], r[i+1] = map(int, input().split())
5
6
7 def dfs(x, h):
```

```

8     global ans
9     if h == len(ans):
10         ans.append(x)
11     else:
12         ans[h] = x
13         if l[x]+1:
14             dfs(l[x], h+1)
15         if r[x]+1:
16             dfs(r[x], h+1)
17
18
19 dfs(1, 0)
20 print(*ans)
21

```

代码运行截图

#45032348提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

l, r = {}, {}
ans = []
for i in range(int(input())):
    l[i+1], r[i+1] = map(int, input().split())

def dfs(x, h):
    global ans
    if h == len(ans):
        ans.append(x)
    else:
        ans[h] = x
        if l[x]+1:
            dfs(l[x], h+1)
        if r[x]+1:
            dfs(r[x], h+1)

dfs(1, 0)
print(*ans)

```

基本信息

#: 45032348

题目: 22485

提交人: 23n2300011505(12号娱乐选手)

内存: 3772kB

时间: 20ms

语言: Python3

提交时间: 2024-05-21 12:56:59

## 28203: 【模板】单调栈

<http://cs101.openjudge.cn/practice/28203/>

思路:

用一个辅助栈，存储还未找到比它更大的数的那些元素，以及这些元素的索引

当每次有新的数进来，就从栈顶比较，弹出所有比当前数更小的元素，并且在它们原来的索引处存储当前新数的索引

辅助栈初始化为全0，这样没找到的数自动就是0了

代码

```
1 n = int(input())
2 s = []
3 ans = [0]*n
4 for i, a in enumerate(map(int, input().split())):
5     while s and s[-1][0] < a:
6         ans[s.pop()[1]] = i+1
7     s.append((a, i))
8 print(*ans)
9
```

代码运行截图

#45032435提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: Accepted

源代码

```
n = int(input())
s = []
ans = [0]*n
for i, a in enumerate(map(int, input().split())):
    while s and s[-1][0] < a:
        ans[s.pop()[1]] = i+1
    s.append((a, i))
print(*ans)
```

基本信息

#: 45032435

题目: 28203

提交人: 23n2300011505(12号娱乐选手)

内存: 359964kB

时间: 3139ms

语言: Python3

提交时间: 2024-05-21 13:12:53

## 09202: 舰队、海域出击!

<http://cs101.openjudge.cn/practice/09202/>

思路:

有向图判断有无环，即判断是否存在这样的点，能够走回自己

进行dfs，同时存储当前能够到达这个点的所有点（开始新的dfs后，不需要更新，因为我们只关心是不是有环），当下一个点已经走过，但是这个点出现在存储之中，说明这个点能走回自己，就判定为有环

代码

```
1 def dfs(x, d):
2     global g, next, f
3     for a in next[x]:
4         if g[a]:
5             g[a] = 0
6             dfs(a, d | {a})
7         else:
8             if a in d:
9                 f = 1
```

```
10         if f:
11             return
12
13
14 def ans():
15     global g, next, f
16     f = 0
17     n, m = map(int, input().split())
18     g = [1]*n
19     next = {i: set() for i in range(n)}
20     for i in range(m):
21         a, b = map(int, input().split())
22         next[a-1].add(b-1)
23     for i in range(n):
24         if g[i]:
25             g[i] = 0
26             dfs(i, {i})
27             if f:
28                 return 'Yes'
29     return 'No'
30
31
32 for _ in range(int(input())):
33     print(ans())
34
```

代码运行截图

状态: Accepted

源代码

```
def dfs(x, d):
    global g, next, f
    for a in next[x]:
        if g[a]:
            g[a] = 0
            dfs(a, d | {a})
        else:
            if a in d:
                f = 1
            if f:
                return

def ans():
    global g, next, f
    f = 0
    n, m = map(int, input().split())
    g = [1]*n
    next = {i: set() for i in range(n)}
    for i in range(m):
        a, b = map(int, input().split())
        next[a-1].add(b-1)
    for i in range(n):
        if g[i]:
            g[i] = 0
            dfs(i, {i})
            if f:
                return 'Yes'
    return 'No'

for _ in range(int(input())):
    print(ans())
```

基本信息

#: 45032704

题目: 09202

提交人: 23n2300011505(12号娱乐选手)

内存: 107172kB

时间: 4114ms

语言: Python3

提交时间: 2024-05-21 13:48:41

## 04135: 月度开销

<http://cs101.openjudge.cn/practice/04135/>

思路:

划分方法是否合法，是可以通过大小判断出来的，因此这个问题可以用二分法解决，搜索到满足要求的最小的月度开销即为答案

代码

```
1 def check(l, m, k):
2     count = 0
3     for a in l:
4         if count+a > k:
5             m -= 1
6             count = 0
7         count += a
8     return m >= 1
9
10
11 n, m = map(int, input().split())
```

```

12 l = []
13 for _ in range(n):
14     l.append(int(input()))
15 left, right = max(l)-1, sum(l)
16 while right-left > 1:
17     middle = (left+right)//2
18     if check(l, m, middle):
19         right = middle
20     else:
21         left = middle
22 print(right)
23

```

代码运行截图

#44083998提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

def check(l, m, k):
    count = 0
    for a in l:
        if count+a > k:
            m -= 1
            count = 0
        count += a
    return m >= 1

n, m = map(int, input().split())
l = []
for _ in range(n):
    l.append(int(input()))
left, right = max(l)-1, sum(l)
while right-left > 1:
    middle = (left+right)//2
    if check(l, m, middle):
        right = middle
    else:
        left = middle
print(right)

```

基本信息

#: 44083998  
 题目: 04135  
 提交人: 23n2300011505(12号娱乐选手)  
 内存: 7444kB  
 时间: 395ms  
 语言: Python3  
 提交时间: 2024-03-05 22:39:11

## 07735: 道路

<http://cs101.openjudge.cn/practice/07735/>

思路:

要用到一种特殊的Dijkstra, 因为有可能出现路更长, 但是花费更小的可行情况

不能走的条件是走了重复路线, 一个很简单的判别方法是记录当前已经走过的道路数量, 当超过N-1则一定走了重复的路

一开始我用路程或者花费变小来剪枝, 这样会错, 会漏掉情况, 而上述剪枝方法由于和Dijkstra连用, 不需要担心路径变长的问题

## 代码

```
1 from heapq import heappop, heappush
2 K = int(input())
3 N = int(input())
4 g = [[] for _ in range(N)]
5 for _ in range(int(input())):
6     S, D, L, T = map(int, input().split())
7     g[S-1].append((D-1, L, T))
8 q = []
9 heappush(q, (0, 0, 0, 0))
10 while q:
11     x, y, n, a = heappop(q)
12     if a == N-1:
13         print(x)
14         exit()
15     for b, l, t in g[a]:
16         if n+1 < N and y+t <= K:
17             heappush(q, (x+l, y+t, n+1, b))
18 print(-1)
19
```

## 代码运行截图

### #45035923提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

#### 源代码

```
from heapq import heappop, heappush
K = int(input())
N = int(input())
g = [[] for _ in range(N)]
for _ in range(int(input())):
    S, D, L, T = map(int, input().split())
    g[S-1].append((D-1, L, T))
q = []
heappush(q, (0, 0, 0, 0))
while q:
    x, y, n, a = heappop(q)
    if a == N-1:
        print(x)
        exit()
    for b, l, t in g[a]:
        if n+1 < N and y+t <= K:
            heappush(q, (x+l, y+t, n+1, b))
print(-1)
```

#### 基本信息

#: 45035923  
题目: 07735  
提交人: 23n2300011505(12号娱乐选手)  
内存: 6932kB  
时间: 48ms  
语言: Python3  
提交时间: 2024-05-21 19:14:41

## 01182: 食物链

<http://cs101.openjudge.cn/practice/01182/>

思路:

并查集，但要注意每个元素存三次，分别表示这个元素作为A/B/C时的情况

当输入x和y是同类，先判断，如果是真的，则将三种情况都归为同类

当输入x吃y，先判断，如果是真的，则同时有y吃x的天敌，x的天敌吃x

代码

```
1 def F(x):
2     global p
3     if p[x] != x:
4         p[x] = F(p[x])
5     return p[x]
6
7
8 def U(x, y):
9     global p
10    x = F(x)
11    y = F(y)
12    p[x] = y
13
14
15 n, k = map(int, input().split())
16 p = [i for i in range(3*n+1)]
17 ans = 0
18 for _ in range(k):
19     d, x, y = map(int, input().split())
20     if x > n or y > n:
21         ans += 1
22         continue
23     if d == 1:
24         if F(x+n) == F(y) or F(x+2*n) == F(y):
25             ans += 1
26             continue
27         U(x, y)
28         U(x+n, y+n)
29         U(x+2*n, y+2*n)
30     else:
31         if F(x) == F(y) or F(x+2*n) == F(y):
32             ans += 1
33             continue
34         U(x, y+2*n)
35         U(x+n, y)
36         U(x+2*n, y+n)
37 print(ans)
38
```

代码运行截图



状态: [Accepted](#)

源代码

```
def F(x):
    global p
    if p[x] != x:
        p[x] = F(p[x])
    return p[x]

def U(x, y):
    global p
    x = F(x)
    y = F(y)
    p[x] = y

n, k = map(int, input().split())
p = [i for i in range(3*n+1)]
ans = 0
for _ in range(k):
    d, x, y = map(int, input().split())
    if x > n or y > n:
        ans += 1
        continue
    if d == 1:
        if F(x+n) == F(y) or F(x+2*n) == F(y):
            ans += 1
            continue
        U(x, y)
        U(x+n, y+n)
        U(x+2*n, y+2*n)
    else:
        if F(x) == F(y) or F(x+2*n) == F(y):
            ans += 1
            continue
        U(x, y+2*n)
        U(x+n, y)
        U(x+2*n, y+n)
print(ans)
```

基本信息

#: 45037919  
题目: 01182  
提交人: 23n2300011505(12号娱乐选手)  
内存: 9480kB  
时间: 552ms  
语言: Python3  
提交时间: 2024-05-21 21:49:37

## 2. 学习总结和收获

这次的作业覆盖面很广，有树、栈、图、二分查找、Dijkstra和并查集，有几题都不简单，是很好的复习

不知不觉，数算已经快学完了，也断断续续刷了挺多题，明显感觉到代码能力提升

笔试还有一些知识盲区，要好好补补了