

# Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Compiled by 夏天明 元培学院

## 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 10 | 22H2

Python编程环境: Spyder IDE 5.4.3 | Python 3.11.4 64-bit

## 1. 题目

### 22485: 升空的焰火, 从侧面看

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

思路: 直接取层序遍历每层最后一个节点即可。善用列表推导式

代码

```
N = int(input())
child = [[int(i)-1 for i in input().split()] for j in range(N)]
this, ans = [0], [1]
while this := [i for p in this for i in child[p] if i != -2]:
    ans.append(this[-1]+1)
print(*ans)
```

代码运行截图

状态: Accepted

源代码

```
N = int(input())
child = [[int(i)-1 for i in input().split()] for j in range(N)]
this, ans = [0], [1]
while this := [i for p in this for i in child[p] if i != -2]:
    ans.append(this[-1]+1)
print(*ans)
```

基本信息

#: 45033221  
题目: 22485  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 3672kB  
时间: 20ms  
语言: Python3  
提交时间: 2024-05-21 15:03:51

## 28203: 【模板】单调栈

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

思路: 单调栈模板题

代码

```
n = int(input())
a = [int(i) for i in input().split()]
f = [0]*n
stack = []
for i in range(n):
    while stack and a[stack[-1]] < a[i]:
        f[stack.pop()] = i+1
    stack.append(i)
print(*f)
```

代码运行截图

状态: Accepted

源代码

```
n = int(input())
a = [int(i) for i in input().split()]
f = [0]*n
stack = []
for i in range(n):
    while stack and a[stack[-1]] < a[i]:
        f[stack.pop()] = i+1
    stack.append(i)
print(*f)
```

基本信息

#: 45034331  
题目: 28203  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 372724kB  
时间: 3048ms  
语言: Python3  
提交时间: 2024-05-21 16:40:32

## 09202: 舰队、海域出击!

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

思路：反复剔除零入度节点，如果最后有节点未能被剔除那一定有环。剔除的顺序不作要求，这里简单使用列表pop的方式进行。

代码

```
for o in range(int(input())):
    N, M = map(int, input().split())
    graph = [[] for i in range(N)]
    indeg = [0]*N
    for o in range(M):
        x, y = map(int, input().split())
        graph[x-1].append(y-1)
        indeg[y-1] += 1
    zero = [i for i in range(N) if indeg[i] == 0]
    used = [0]*N
    while zero:
        p = zero.pop()
        used[p] = 1
        for nex in graph[p]:
            indeg[nex] -= 1
            if indeg[nex] == 0:
                zero.append(nex)
    print('No' if all(used) else 'Yes')
```

代码运行截图

状态: Accepted

源代码

```
for o in range(int(input())):
    N, M = map(int, input().split())
    graph = [[] for i in range(N)]
    indeg = [0]*N
    for o in range(M):
        x, y = map(int, input().split())
        graph[x-1].append(y-1)
        indeg[y-1] += 1
    zero = [i for i in range(N) if indeg[i] == 0]
    used = [0]*N
    while zero:
        p = zero.pop()
        used[p] = 1
        for nex in graph[p]:
            indeg[nex] -= 1
            if indeg[nex] == 0:
                zero.append(nex)
    print('No' if all(used) else 'Yes')
```

基本信息

#: 45040649  
题目: 09202  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 37788kB  
时间: 3841ms  
语言: Python3  
提交时间: 2024-05-22 11:01:50

## 04135: 月度开销

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

思路：最大值的最小可能值，使用二分查找，定义一个判断尝试的最大值是否可能的函数reachable，他作用在全体可能的最大值（从max(cost)到sum(cost)）相当于一个从0到1的升序列表，然后对1进行左二分查找。这里套用了cheat\_sheet里的懒二分函数

代码

```
def lazybisect_left(start, end, lazylist_func, x):
    l, r = start, end
    while l+1 < r:
        mid = (l+r)//2
        if lazylist_func(mid) >= x:
            r = mid
        else:
            l = mid+1
    return l if lazylist_func(l) == x else r

def reachable(sup):
    tmp = 0
    res = 0
    for c in cost:
        tmp += c
        if tmp > sup:
            res += 1
            tmp = c
```

```

    res += tmp != 0
    return res <= M

N, M = map(int, input().split())
cost = [int(input()) for i in range(N)]
print(lazybisect_left(max(cost), sum(cost), reachable, 1))

```

代码运行截图

#45041346提交状态

[查看](#) [提交](#) [统计](#) [提问](#)

状态: **Accepted**

源代码

```

def lazybisect_left(start, end, lazylist_func, x):
    l, r = start, end
    while l+1 < r:
        mid = (l+r)//2
        if lazylist_func(mid) >= x:
            r = mid
        else:
            l = mid+1
    return l if lazylist_func(l) == x else r

def reachable(sup):
    tmp = 0
    res = 0
    for c in cost:
        tmp += c
        if tmp > sup:
            res += 1
            tmp = c
    res += tmp != 0
    return res <= M

N, M = map(int, input().split())
cost = [int(input()) for i in range(N)]
print(lazybisect_left(max(cost), sum(cost), reachable, 1))

```

基本信息

#: 45041346  
 题目: 04135  
 提交人: 23n2300017735(夏天明  
 BrightSummer)  
 内存: 7932kB  
 时间: 307ms  
 语言: Python3  
 提交时间: 2024-05-22 12:18:54

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 07735: 道路

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

思路: dijk, 但是需要额外存一个参数来计算路费, 在入堆的时候, 如果算出来的路费不超过该节点visited过的最高路费, 就舍弃。

代码

```

from heapq import heappop, heappush

K = int(input())
N = int(input())
road = [[] for i in range(N)]
for o in range(int(input())):

```

```

a, b, d, t = map(int, input().split())
road[a-1].append((b-1, d, t))
q = [(0, 0, K)]
visited = [-1]*N
while q:
    distance, p, money = heappop(q)
    if p == N-1:
        print(distance)
        break
    visited[p] = money
    for nex, d, t in road[p]:
        if visited[nex] < money-t:
            heappush(q, (distance+d, nex, money-t))
    else:
        print(-1)

```

代码运行截图

#45045106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())
road = [[] for i in range(N)]
for o in range(int(input())):
    a, b, d, t = map(int, input().split())
    road[a-1].append((b-1, d, t))
q = [(0, 0, K)]
visited = [-1]*N
while q:
    distance, p, money = heappop(q)
    if p == N-1:
        print(distance)
        break
    visited[p] = money
    for nex, d, t in road[p]:
        if visited[nex] < money-t:
            heappush(q, (distance+d, nex, money-t))
    else:
        print(-1)

```

基本信息

#: 45045106  
 题目: 07735  
 提交人: 23n2300017735(夏天明  
 BrightSummer)  
 内存: 4460kB  
 时间: 46ms  
 语言: Python3  
 提交时间: 2024-05-22 16:39:04

©2002-2022 POJ 京ICP备20010980号-1

[English](#) [帮助](#) [关于](#)

## 01182: 食物链

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

思路: 每个动物存三个节点, 维护并查集。用等号表示在同一集合, 则记a与b同类:  $a_1=b_1, a_2=b_2, a_3=b_3$ ; a被b吃:  $a_1=b_2, a_2=b_3, a_3=b_1$ ; a吃b:  $a_1=b_3, a_2=b_1, a_3=b_2$ 。

这样可以维护两个函数isType和setType, 分别表示查询a和b的关系、设定a和b的关系, 其中关系参数用相位差表示。

对于每句话，例如“A与B同类”，如果其相斥命题“A与B吃或被吃（即相位差 $t=1$ 或 $2$ ）”有一个可以被isType查询为成立，则这句话就是假话，反之则为真话，使用setType将A与B相位差设为0。

代码

```
class Dsu:
    def __init__(self, size):
        self.pa = list(range(size))

    def find(self, x):
        if self.pa[x] != x:
            self.pa[x] = self.find(self.pa[x])
        return self.pa[x]

    def union(self, x, y):
        self.pa[self.find(x)] = self.find(y)

def isType(X, Y, t):
    return dsu.find(3*(X-1)) == dsu.find(3*(Y-1)+t)

def setType(X, Y, t):
    for i in range(3):
        dsu.union(3*(X-1)+i, 3*(Y-1)+(i+t)%3)

N, K = map(int, input().split())
dsu = Dsu(3*N)
ans = 0
for o in range(K):
    D, X, Y = map(int, input().split())
    if X > N or Y > N:
        ans += 1
        continue
    if D == 1:
        if any(isType(X, Y, t) for t in [1,2]):
            ans += 1
        else:
            setType(X, Y, 0)
    else:
        if any(isType(X, Y, t) for t in [0,1]):
            ans += 1
        else:
            setType(X, Y, 2)
print(ans)
```

代码运行截图

状态: [Accepted](#)

源代码

```
class Dsu:
    def __init__(self, size):
        self.pa = list(range(size))

    def find(self, x):
        if self.pa[x] != x:
            self.pa[x] = self.find(self.pa[x])
        return self.pa[x]

    def union(self, x, y):
        self.pa[self.find(x)] = self.find(y)

def isType(X, Y, t):
    return dsu.find(3*(X-1)) == dsu.find(3*(Y-1)+t)

def setType(X, Y, t):
    for i in range(3):
        dsu.union(3*(X-1)+i, 3*(Y-1)+(i+t)%3)

N, K = map(int, input().split())
dsu = Dsu(3*N)
ans = 0
for o in range(K):
```

基本信息

#: 45045423  
题目: 01182  
提交人: 23n2300017735(夏天明  
BrightSummer)  
内存: 9416kB  
时间: 826ms  
语言: Python3  
提交时间: 2024-05-22 16:54:44

## 2. 学习总结和收获

有很多之前每日选做的题，整体上难度适中，但涉及图搜索、二分搜索的应用题，细节会比较多，需要一点点debug。

食物链题也是之前每日选做题，比较有趣，很多时候并查集会以各种变式出现，选择把一个对象用多个节点共同维护（比如这里面用三个节点维护一个动物）很多时候是一种不错的思路。