Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied by Xinjie Song, Phy

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 11 22H2

Python编程环境：PyCharm 2023.2 (Community Edition)

C/C++编程环境：g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 题目

22485: 升空的焰火，从侧面看

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

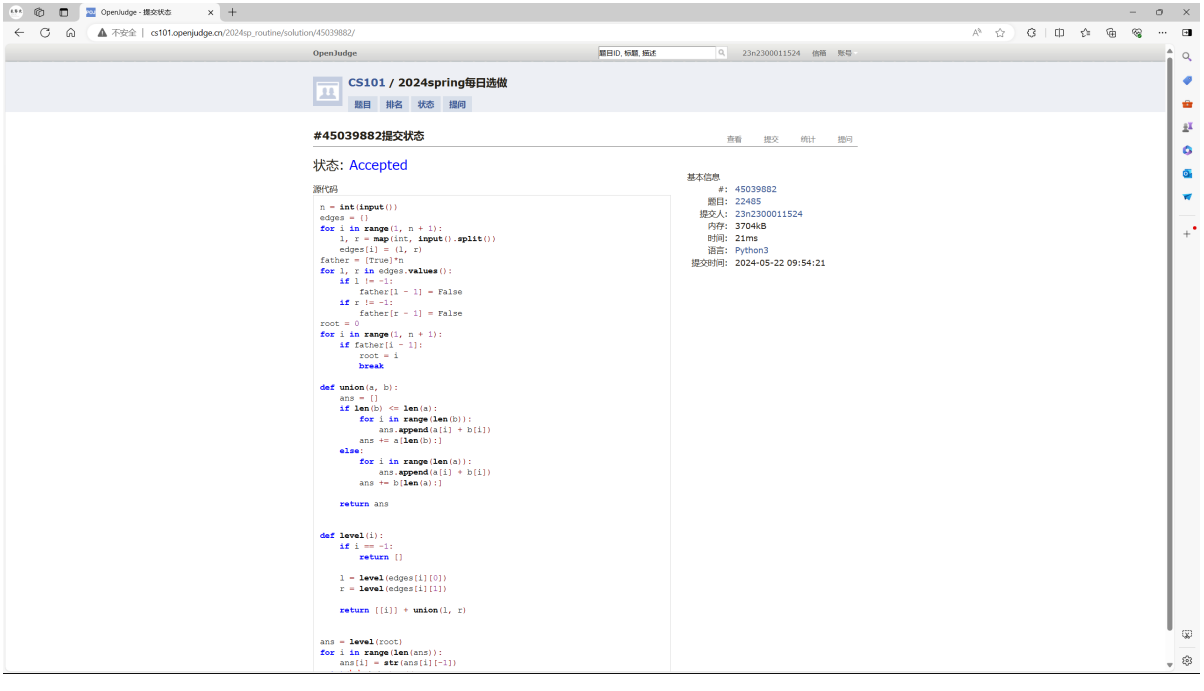
思路：按层次遍历取每一层的末尾

代码

```
n = int(input())
edges = {}
for i in range(1, n + 1):
    l, r = map(int, input().split())
    edges[i] = (l, r)
father = [True]*n
for l, r in edges.values():
    if l != -1:
        father[l - 1] = False
    if r != -1:
        father[r - 1] = False
root = 0
for i in range(1, n + 1):
    if father[i - 1]:
```

```
        root = i
        break

def union(a, b):
    ans = []
    if len(b) <= len(a):
        for i in range(len(b)):
            ans.append(a[i] + b[i])
        ans += a[len(b):]
    else:
        for i in range(len(a)):
            ans.append(a[i] + b[i])
        ans += b[len(a):]

    return ans


def level(i):
    if i == -1:
        return []

    l = level(edges[i][0])
    r = level(edges[i][1])

    return [[i]] + union(l, r)


ans = level(root)
for i in range(len(ans)):
    ans[i] = str(ans[i][-1])
print(' '.join(ans))
```

代码运行截图



28203:【模板】单调栈

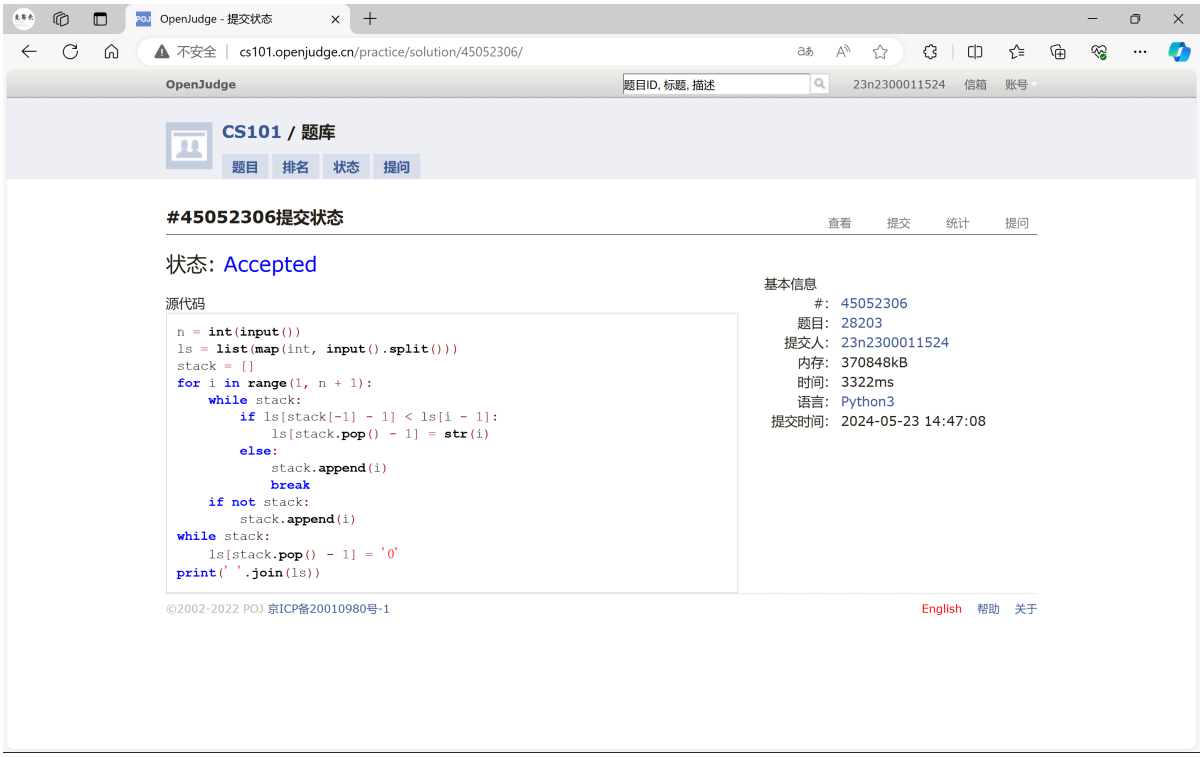
http://cs101.openjudge.cn/practice/28203/

思路：感觉单调栈挺妙的；本来写了个反向的结果超内存了，看了题解改成了正向的，感觉正向的难想一些，脑子需要多转几圈

代码

```
n = int(input())
ls = list(map(int, input().split()))
stack = []
for i in range(1, n + 1):
    while stack:
        if ls[stack[-1] - 1] < ls[i - 1]:
            ls[stack.pop() - 1] = str(i)
        else:
            stack.append(i)
            break
    if not stack:
        stack.append(i)
while stack:
    ls[stack.pop() - 1] = '0'
print(' '.join(ls))
```

代码运行截图



09202: 舰队、海域出击!

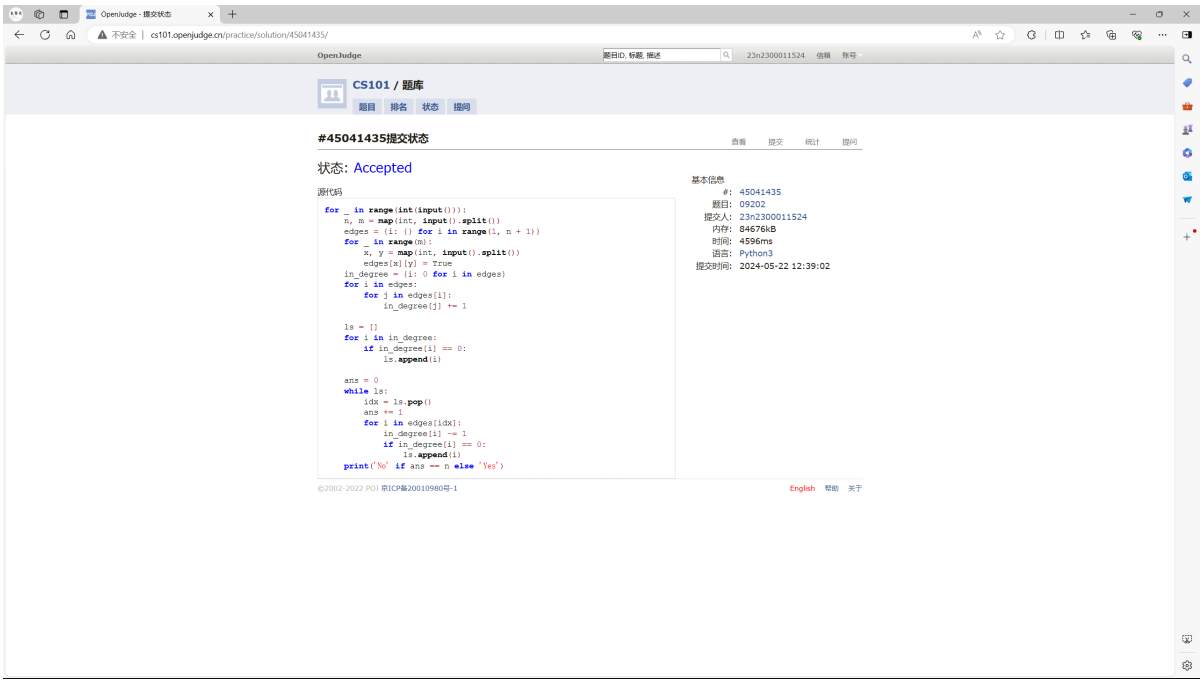
http://cs101.openjudge.cn/practice/09202/

思路：拓扑排序

代码

```
for _ in range(int(input())):
    n, m = map(int, input().split())
    edges = {i: {} for i in range(1, n + 1)}
    for _ in range(m):
        x, y = map(int, input().split())
        edges[x][y] = True
    in_degree = {i: 0 for i in edges}
    for i in edges:
        for j in edges[i]:
            in_degree[j] += 1

    ls = []
    for i in in_degree:
        if in_degree[i] == 0:
            ls.append(i)

    ans = 0
    while ls:
        idx = ls.pop()
        ans += 1
        for i in edges[idx]:
            in_degree[i] -= 1
            if in_degree[i] == 0:
                ls.append(i)
    print('No' if ans == n else 'Yes')#
```

代码运行截图



04135: 月度开销

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

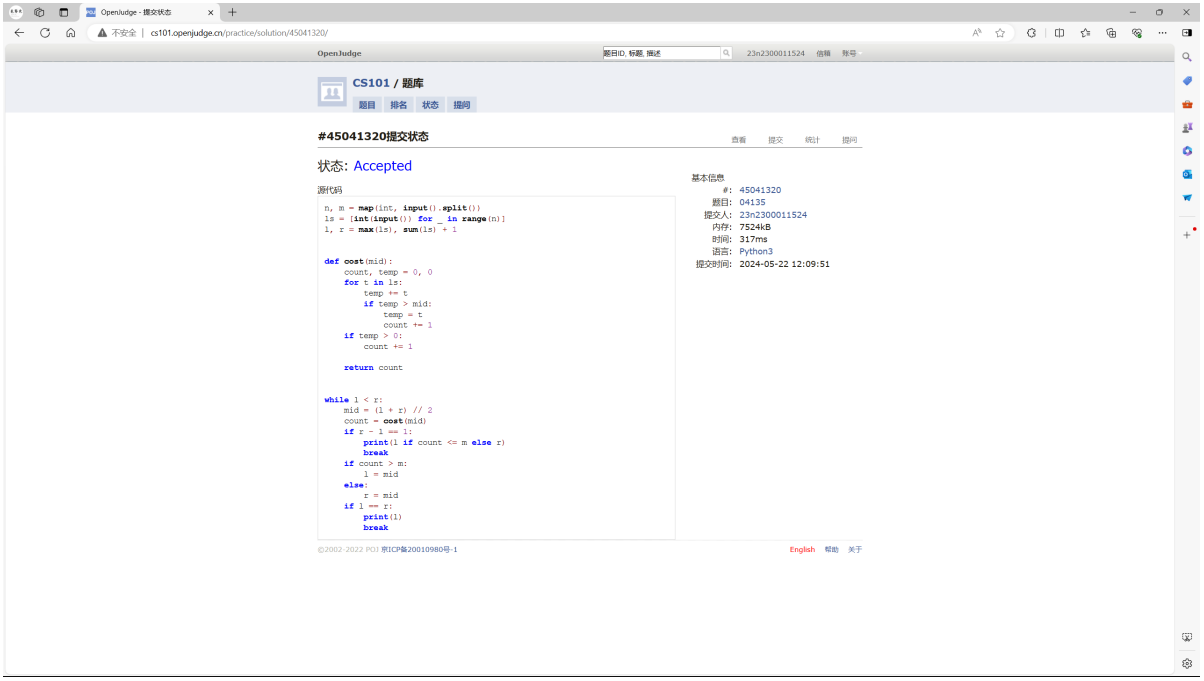
思路：二分

代码

```
n, m = map(int, input().split())
ls = [int(input()) for _ in range(n)]
l, r = max(ls), sum(ls) + 1


def cost(mid):
    count, temp = 0, 0
    for t in ls:
        temp += t
        if temp > mid:
            temp = t
            count += 1
    if temp > 0:
        count += 1

    return count


while l < r:
    mid = (l + r) // 2
    count = cost(mid)
    if r - l == 1:
        print(l if count <= m else r)
        break
```

```
    if count > m:
        l = mid
    else:
        r = mid
    if l == r:
        print(l)
        break
```

代码运行截图



07735: 道路

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

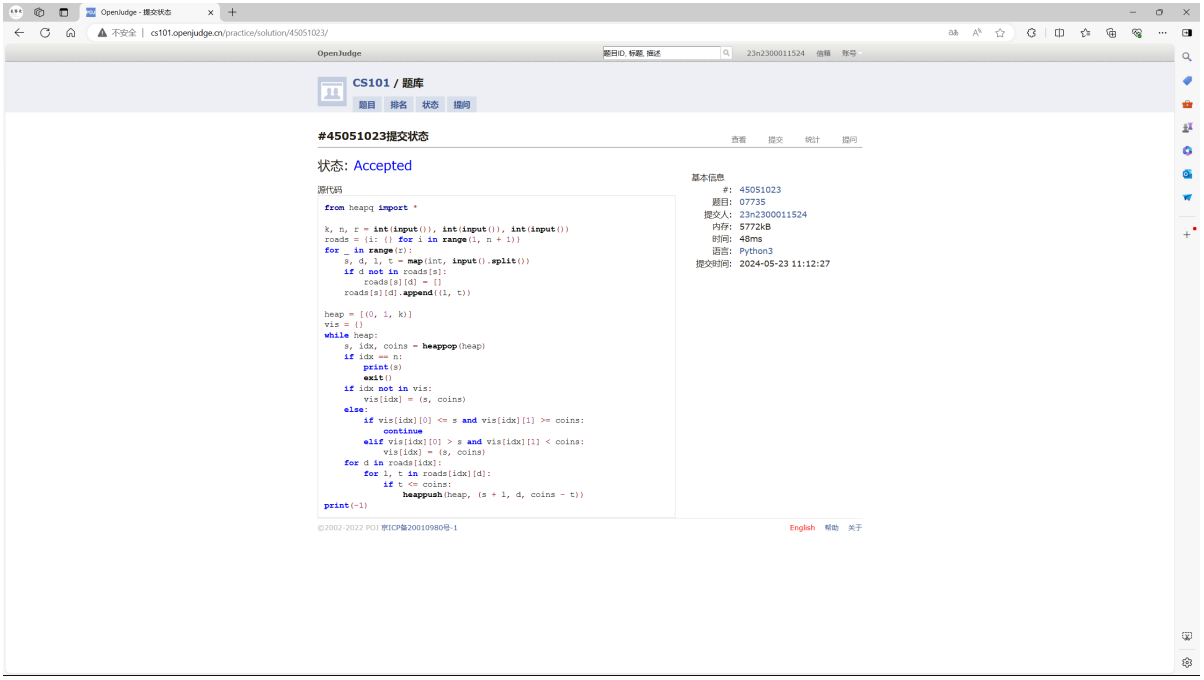
思路：bfs+heap（dji什么不会拼）

代码

```
from heapq import *

k, n, r = int(input()), int(input()), int(input())
roads = {i: {} for i in range(1, n + 1)}
for _ in range(r):
    s, d, l, t = map(int, input().split())
    if d not in roads[s]:
        roads[s][d] = []
    roads[s][d].append((l, t))

heap = [(0, 1, k)]
vis = {}
while heap:
```

```
    s, idx, coins = heappop(heap)
    if idx == n:
        print(s)
        exit()
    if idx not in vis:
        vis[idx] = (s, coins)
    else:
        if vis[idx][0] <= s and vis[idx][1] >= coins:
            continue
        elif vis[idx][0] > s and vis[idx][1] < coins:
            vis[idx] = (s, coins)
    for d in roads[idx]:
        for l, t in roads[idx][d]:
            if t <= coins:
                heappush(heap, (s + l, d, coins - t))
print(-1)
```

代码运行截图



01182: 食物链

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

思路：本题用时较长，经过多次思绪打结重写后，找到的一种思路为：为每一个动物分配一个环形食物链`Circle`类，每个动物为自己的食物环的A种动物，并且该食物环的根为该动物，若`D=1`，两个动物所在食物环的根不同，那么经过适当旋转后，合并两个食物环；两个动物所在的食物环相同，但两个动物种类不同，则为假话。若`D=2`，两个动物所在食物环的根不同，那么经过适当旋转后，合并两个食物环；两个动物所在的食物环相同，但两个动物种类不满足A吃B，B吃C，C吃A，则为假话。为什么没有为每一个动物分配一个动物类？因为处理很多未知种类的动物之间的吃与被吃关系让人头大。

代码

```
class Circle:
    def __init__(self, root):
        self.kinds = {root: 0}
        self.root = root

    def join(self, circle, rotate):
        for idx, kind in circle.kinds.items():
            self.kinds[idx] = (kind - rotate) % 3
            ls[idx] = self.root


n, k = map(int, input().split())
ls = [i for i in range(n)]
circles = [Circle(i) for i in range(n)]
ans = 0

for _ in range(k):
    d, x, y = map(int, input().split())
    if x > n or y > n or (d == 2 and x == y):
        ans += 1
        continue
    x -= 1
    y -= 1

    idx_x = circles[ls[x]].kinds[x]
    idx_y = circles[ls[y]].kinds[y]
    if d == 1:
        if ls[x] != ls[y]:
            circles[ls[x]].join(circles[ls[y]], (idx_y - idx_x) % 3)
        else:
            if idx_x != idx_y:
                ans += 1
        continue

    if ls[x] != ls[y]:
        circles[ls[x]].join(circles[ls[y]], (idx_y - idx_x - 1) % 3)
    elif (idx_y - idx_x - 1) % 3 != 0:
        ans += 1

print(ans)
```

代码运行截图

2. 学习总结和收获

感觉本次作业难度其他<单调栈<食物链，单调栈终于是学会了，担心下次遇到类似食物链的需要非常规思路（找到动物类背后的食物链类作为处理对象）的题能快速想到简便思路。