Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied by 同学的姓名、院系

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

（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

22485: 升空的焰火，从侧面看

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

思路：
想用字典做，但是没想好怎么确定层级，参考答案

代码

```
from collections import deque

def right_view(n, tree):
    queue = deque([(1, tree[1])])  # start with root node
    right_view = []

    while queue:
        level_size = len(queue)
        for i in range(level_size):
            node, children = queue.popleft()
            if children[0] != -1:
                queue.append((children[0], tree[children[0]]))
            if children[1] != -1:
                queue.append((children[1], tree[children[1]]))
        right_view.append(node)

    return right_view

n = int(input())
tree = {1: [-1, -1] for _ in range(n+1)}  # initialize tree with -1s
for i in range(1, n+1):
    left, right = map(int, input().split())
    tree[i] = [left, right]

result = right_view(n, tree)
print(' '.join(map(str, result)))
```

代码运行截图 （至少包含有"Accepted"）

28203:【模板】单调栈

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

思路：

有点看不懂，我会再看看

代码

```
n = int(input())
a = list(map(int, input().split()))
stack = []

for i in range(n):
    while stack and a[stack[-1]] < a[i]:
        a[stack.pop()] = i + 1


    stack.append(i)

while stack:
    a[stack[-1]] = 0
    stack.pop()

print(*a)
```

代码运行截图 （至少包含有"Accepted"）

09202: 舰队、海域出击！

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

思路：

代码

```
from collections import deque,defaultdict
def topo_sort(graph):
    in_degree={u:0 for u in range(1,n+1)}
    for u in graph:
        for v in graph[u]:
            in_degree[v]+=1
    q=deque([u for u in in_degree if in_degree[u]==0])
    topo_order=[]
    while q:
        u=q.popleft()
        topo_order.append(u)
        for v in graph[u]:
            in_degree[v]-=1
            if in_degree[v]==0:
                q.append(v)
    if len(topo_order)!=len(graph):
        return 'Yes'
    return 'No'
for _ in range(int(input())):
    n,m=map(int,input().split())
    graph=defaultdict(list)
    for _ in range(m):
        u,v=map(int,input().split())
        graph[u].append(v)
    print(topo_sort(graph))
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）
alt text

04135: 月度开销

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

思路：
不是很懂，参考答案

代码

```python
n,m = map(int, input().split())
expenditure = []
for _ in range(n):
    expenditure.append(int(input()))

def check(x):
    num, s = 1, 0
    for i in range(n):
        if s + expenditure[i] > x:
            s = expenditure[i]
            num += 1
        else:
            s += expenditure[i]

    return [False, True][num > m]

# https://github.com/python/cpython/blob/main/Lib/bisect.py
lo = max(expenditure)
# hi = sum(expenditure)
hi = sum(expenditure) + 1
ans = 1
while lo < hi:
    mid = (lo + hi) // 2
    if check(mid):      # 返回True，是因为num>m，是确定不合适
        lo = mid + 1    # 所以lo可以置为 mid + 1。
    else:
        ans = mid    # 如果num==m，mid可能是答案
        hi = mid

#print(lo)
print(ans)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）

alt text

07735: 道路

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

思路：
剪枝问题，我还得多看一下

代码

```
import heapq

def dijkstra(g):
    while pq:
        dist,node,fee = heapq.heappop(pq)
        if node == n-1 :
            return dist
        for nei,w,f in g[node]:
            n_dist = dist + w
            n_fee = fee + f
            if n_fee <= k:
                dists[nei] = n_dist
                heapq.heappush(pq,(n_dist,nei,n_fee))
    return -1

k,n,r = int(input()),int(input()),int(input())
g = [[] for _ in range(n)]
for i in range(r):
    s,d,l,t = map(int,input().split())
    g[s-1].append((d-1,l,t)) #node,dist,fee

pq = [(0,0,0)] #dist,node,fee
dists = [float('inf')] * n
dists[0] = 0
spend = 0

result = dijkstra(g)
print(result)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）
alt text

01182: 食物链

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

思路：
确实不会做

代码

```
# 并查集，https://zhuanlan.zhihu.com/p/93647900/
'''
我们设[0,n)区间表示同类，[n,2*n)区间表示x吃的动物，[2*n,3*n)表示吃x的动物。

如果是关系1：
    将y和x合并。将y吃的与x吃的合并。将吃y的和吃x的合并。
如果是关系2：
    将y和x吃的合并。将吃y的与x合并。将y吃的与吃x的合并。
原文链接：https://blog.csdn.net/qq_34594236/article/details/72587829
'''
# p = [0]*150001

def find(x):        # 并查集查询
    if p[x] == x:
        return x
    else:
        p[x] = find(p[x])        # 父节点设为根节点。目的是路径压缩。
        return p[x]

n,k = map(int, input().split())

p = [0]*(3*n + 1)
for i in range(3*n+1):        #并查集初始化
    p[i] = i

ans = 0
for _ in range(k):
    a,x,y = map(int, input().split())
    if x>n or y>n:
        ans += 1; continue

    if a==1:
        if find(x+n)==find(y) or find(y+n)==find(x):
            ans += 1; continue

        # 合并
        p[find(x)] = find(y)
        p[find(x+n)] = find(y+n)
        p[find(x+2*n)] = find(y+2*n)
```

```
    else:
        if find(x)==find(y) or find(y+n)==find(x):
            ans += 1; continue
        p[find(x+n)] = find(y)
        p[find(y+2*n)] = find(x)
        p[find(x+2*n)] = find(y+n)

print(ans)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）
alt text

2. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“2024spring每日选做”、CF、LeetCode、洛谷等网站题目。
最大的感受是模板题也挺难的，这一周好好复习，争取让我ac3吧