Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied by 夏天、生命科学学院

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

(请改为同学的操作系统、编程环境等)

操作系统: Windows 10 家庭版

Python编程环境: Spyder (python 3.11)

1. 题目

22485: 升空的焰火,从侧面看

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

思路:模板题,建树 层次遍历,只需额外记录每一层的节点个数n,进行n次节点的子节点入队的循环后,把最后 pop出来的节点加入答案列表,即层次遍历的最后一个节点即可

代码

```
from collections import deque
class Treenode:
      def __init__(self,value):
           self.value=value
           self.left=None
           self.right=None
N=int(input())
nodes={i:Treenode(i)for i in range(1,N+1)}
for i in range(1,N+1):
     a,b=map(int,input().split())
      if a!=-1:
           nodes[i].left=nodes[a]
      if b! = -1:
           nodes[i].right=nodes[b]
                                                    状态: Accepted
node=nodes[1]
                                                                                                                    # 45035060
                                                                                                                  题目: 22485
queue=deque([node])
                                                     from collections import deque
                                                                                                                 提交人: 23n2300012289
                                                     class Treenode:
while queue:
                                                                                                                  内存: 3776kB
                                                       def __init__(self,value):
     self.value=value
     self.left=None
      level_size=len(queue)
                                                                                                                 时间: 22ms
                                                                                                                  语言: Python3
      for i in range(level_size):
                                                           self.right=None
                                                                                                               提交时间: 2024-05-21 19:19:26
                                                    N=int(input())
nodes={i:Treenode(i)for i in range(1,N+1)}
for i in range(1,N+1):
           node=queue.popleft()
            if node.left:
                                                     a,b=map(int,input().split())
if a!=-1:
                 queue.append(node.left)
           if node.right:
                                                           nodes[i].left=nodes[a]
                                                       if b!=-1:
                 queue.append(node.right)
                                                           nodes[i].right=nodes[b]
                                                     ans=[]
      ans.append(node.value)
                                                     node=nodes[1]
print(*ans)
                                                     queue=deque([node])
                                                     while queue:
代码运行截图 (至少包含有"Accepted")
                                                        level_size=len(queue)
for i in range(level_size):
                                                           node=queue.popleft()
if node.left:
                                                              queue.append(node.left)
                                                           if node.right:
                                                        queue.append(node.right)
ans.append(node.value)
```

print(*ans

28203:【模板】单调栈

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

思路:由于是找第i个元素之后第一个大于ai的元素的下标,所以就想到了倒着遍历:f(n)一定为0,用栈储存倒着看元素满足严格单调下降的索引,则栈顶的索引对应的元素一定是已遍历序列中最小的元素m;如果当前元素比m小,则当前元素的索引入栈,否则弹出栈顶元素,重复上述操作;每个答案都是当时栈顶元素+1(栈非空时)

代码

```
n=int(input())
numbers=list(map(int,input().split()))
ans=[0]*n
stack=[]
for i in range(n-1,-1,-1):
    while stack and numbers[i]>=numbers[stack[-1]]:
        stack.pop()
    if stack:
        ans[i]=stack[-1]+1
    stack.append(i)
print(*ans)
```

代码运行截图 <mark>(至少包含有"Accepted")</mark> 状态: Accepted

```
基本信息
                                                                              #: 45036395
                                                                            题目: 28203
n=int(input())
                                                                          提交人: 23n2300012289
numbers=list(map(int,input().split()))
                                                                            内存: 359988kB
ans=[0]*n
stack=[]
                                                                            时间: 3096ms
for i in range (n-1,-1,-1):
                                                                            语言: Pvthon3
   while stack and numbers[i]>=numbers[stack[-1]]:
                                                                         提交时间: 2024-05-21 19:54:34
       stack.pop()
   if stack:
       ans[i]=stack[-1]+1
   stack.append(i)
print(*ans)
```

09202: 舰队、海域出击!

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

思路: 模板题,有向图判环,用拓扑排序。

代码

代码运行截图 (AC代码截图,至少包含有"Accepted")

```
from collections import defaultdict, deque
                                                                                                                                                                                                                状态: Accepted
def has_loop(graph):
                   queue=deque([])
                                                                                                                                                                                                                  from collections import defaultdict, deque

def has_loop(graph);

queue-deque();

for i in range(i,N+1);

if in_degrees[i]=0;

queue.append(i)

while queue:

node=queue.popleft()

visited.add(node)
                                                                                                                                                                                                                                                                                                                                                                                               题目: 09202
                   for i in range(1,N+1):
                                                                                                                                                                                                                                                                                                                                                                                           提交人: 23n2300012289
                                                                                                                                                                                                                                                                                                                                                                                              内存: 68572kB
时间: 4010ms
                                       if in_degrees[i]==0:
                                                         queue.append(i)
                                                                                                                                                                                                                                                                                                                                                                                       语言: Python3
提交时间: 2024-05-22 19:38:42
                   while queue:
                                      node=queue.popleft()
                                                                                                                                                                                                                            visited.add(node)
for neighbor in graph[node]:
    in_degrees[neighbor]==1
    if in_degrees[neighbor]==0
    queue.append(neighbor)
return N!=len(visited)
t(innut())
                                       visited.add(node)
                                       for neighbor in graph[node]:
                                                                                                                                                                                                                      return N!=len(visited)
=int(input())
for _ in range(T):
   N,M=map(int,input().split())
   graph=defaultdiot(list)
   in_degrees=[0]*(N+1)
   visited=set()
for an arrangement of the set of the 
                                                           in_degrees[neighbor]-=1
                                                           if in_degrees[neighbor]==0:
                                                                               queue.append(neighbor)
                                                                                                                                                                                                                          visited=set()
for _ in range(M):
    x,y=map(int,input().split())
    graph(X).append(y)
    in_degrees(y)+=1
    if has_loop(graph):
    print('Yes')
    else:
        print('No')
                   return N!=len(visited)
T=int(input())
 for _ in range(T):
                   N,M=map(int,input().split())
                   graph=defaultdict(list)
                   in_degrees=[0]*(N+1)
                   visited=set()
                    for _ in range(M):
                                       x,y=map(int,input().split())
                                       graph[x].append(y)
                                       in_degrees[y]+=1
                    if has_loop(graph):
                                       print('Yes')
                   else:
                                      print('No')
```

04135: 月度开销

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

代码

代码运行截图 (AC代码截图,至少包含有"Accepted")

```
def check(budgets,m,mid):
                                                                 状态: Accepted
      count=0
                                                                 源代码
                                                                                                                                      #: 45048218
      total=0
                                                                                                                                    题目: 04135
                                                                  def check (budgets, m, mid):
      for budget in budgets:
                                                                                                                                   提交人: 23n2300012289
                                                                                                                                    内存: 7484kB
             total+=budget
                                                                     total=0
                                                                     for budget in budgets:
    total+=budget
    if total>mid:
                                                                                                                                    时间: 313ms
             if total>mid:
                                                                                                                                    语言: Python3
                                                                                                                                 提交时间: 2024-05-22 20:35:12
                   count+=1
                                                                            total=budget
                    total=budget
                                                                     if total>0
                                                                     count+=1
return count<=m
      if total>0:
             count+=1
                                                                  def min max cost(budgets, n, m):
    left=max(budgets)
    right=sum(budgets)
    while left-right:
        mid=(left+right)//2
      return count<=m
def min max cost(budgets,n,m):
      left=max(budgets)
                                                                        if check (budgets, m, mid) :
                                                                            right=mid
      right=sum(budgets)
                                                                     else:
left=mid+1
return left
      while left<right:</pre>
             mid=(left+right)//2
                                                                  n, m=map(int, input().split())
                                                                  budgets=[int(input()) for _ in range(n)]
print(min_max_cost(budgets,n,m))
             if check(budgets,m,mid):
                   right=mid
             else:
                    left=mid+1
      return left
n,m=map(int,input().split())
budgets=[int(input())for _ in range(n)]
print(min_max_cost(budgets,n,m))
```

07735: 道路

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

思路:通行费让我想起来了鸣人与佐助的查克拉,于是vi si ted里多加了一个参数。由于不同道路可以有相同的起点和终点,为避免覆盖要用列表记录。

代码

```
from collections import defaultdict
import heapq
K=int(input())
N=int(input())
R=int(input())
graph=defaultdict(list)
for _ in range(R):
    S,D,L,T=map(int,input().split())
    graph[S].append((D,L,T))
visited=set()
start=(0,0,1)
heap=[]
heapq.heappush(heap,start)
while heap:
    total_length,total_charge,current_city=heapq.heappop(heap)
    if (current_city,total_charge) in visited:
        continue
    visited.add((current_city,total_charge))
    if current city==N:
        print(total_length)
        break
    for next_city,next_length,next_charge in graph[current_city]:
        if total_charge+next_charge<=K:</pre>
            heapq.heappush(heap,(total_length+next_length,total_charge+next_
charge,next_city))
                                                  状态: Accepted
if current_city!=N:
                                                                                     基本信息
                                                  源代码
                                                                                        #: 45049308
    print(-1)
```

代码运行截图 (AC代码截图,至少包含有"Accepted")

```
# 45049308
## 45049308
## 145049308
## 145049308
## 145049308
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
## 157735
##
```

01182: 食物链

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

思路: 并查集,但是[0, n)表示x的同类,[n, 2*n)表示x吃的,[2*n, 3*n)表示吃x的,每次说话都进行相应合并,一旦出现假话答案就加1并跳过合并

代码

```
def find(x):
    if parents[x]!=x:
       parents[x]=find(parents[x])
    return parents[x]
N,K=map(int,input().split())
parents=[0]*(3*N+1)
for i in range(3*N+1):
    parents[i]=i
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 find(X+N)==find(Y) or find(Y+N)==find(X):
            continue
        parents[find(X)]=find(Y)
        parents[find(X+N)]=find(Y+N)
        parents[find(X+2*N)]=find(Y+2*N)
    else:
        if find(X) == find(Y) or find(Y+N) == find(X):
            ans+=1
            continue
        parents[find(X+N)]=find(Y)
        parents[find(Y+2*N)]=find(X)
        parents[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")

状态: Accepted

```
基本信息
源代码
                                                                                          #: 45049226
                                                                                        题目: 01182
 def find(x):
                                                                                      提交人: 23n2300012289
     if parents[x]!=x:
                                                                                        内存: 9380kB
          parents[x]=find(parents[x])
 return parents[x]
N,K=map(int,input().split())
                                                                                       时间: 515ms
                                                                                        语言: Python3
                                                                                    提交时间: 2024-05-22 22:22:53
 parents=[0]*(3*N+1)
 for i in range (3*N+1):
     parents[i]=i
 ans=0
 for _ in range(K):
    D, X, Y=map(int, input().split())
      if X>N or Y>N:
          ans+=1
      if D==1:
          if find(X+N) ==find(Y) or find(Y+N) ==find(X):
            ans+=1
          parents[find(X)]=find(Y)
parents[find(X+N)]=find(Y+N)
          parents[find(X+2*N)]=find(Y+2*N)
          if find(X) ==find(Y) or find(Y+N) ==find(X):
              continu
          parents[find(X+N)]=find(Y)
          parents[find(Y+2*N)]=find(X)
          parents[find(X+2*N)]=find(Y+N)
```

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。这次作业涉及知识点还挺全的,二叉树的层次遍历、单调栈、拓扑排序、二分查找、Dijkstra、并查集。嗯,似乎都是模板题,或许可以直接把这次作业当成cheat sheet的一部分(

从昨天的笔试来看自己还是有一些概念和细节不清楚,光死记硬背感觉还是没法提高熟练度,尝试通过做题帮助自己理解这些概念