

Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Compiled by ==田济维 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

(python pycharm)

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:

代码

```
1  #
2  from collections import deque
3  # 第一步建树
4  class Node:
5      def __init__(self, key):
6          self.key = key
7          self.left = None
8          self.right = None
9  N = int(input())
```

```

10 Nodes = [Node(i) for i in range(1,N+1)]
11 for i in range(N):
12     l,r = map(int,input().split())
13     if l!=-1:
14         Nodes[i].left = Nodes[l-1]
15     if r!=-1:
16         Nodes[i].right = Nodes[r-1]
17
18 #第二步按层次遍历树
19 def traverse(tree):
20     que = deque([tree])
21     result = []
22     while que:
23         result.append(que[-1].key)
24         n = len(que)
25         for i in range(n):
26             s = que.popleft()
27             if s.left:
28                 que.append(s.left)
29             if s.right:
30                 que.append(s.right)
31     for x in result:
32         print(x,end = " ")
33 traverse(Nodes[0])
34
35

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

源代码

```
from collections import deque
# 第一步建树
class Node:
    def __init__(self, key):
        self.key = key
        self.left = None
        self.right = None
N = int(input())
Nodes = [Node(i) for i in range(1, N+1)]
for i in range(N):
    l, r = map(int, input().split())
    if l != -1:
        Nodes[i].left = Nodes[l-1]
    if r != -1:
        Nodes[i].right = Nodes[r-1]

# 第二步按层次遍历树
def traverse(tree):
    que = deque([tree])
    result = []
    while que:
        result.append(que[-1].key)
        n = len(que)
        for i in range(n):
            s = que.popleft()
            if s.left:
```

28203: 【模板】单调栈

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

思路:

代码

```
1 #
2 N = int(input())
3 que = list(map(int, input().split()))
4
5 stack = []
6 ff = []
7 for i in range(N-1, -1, -1):
8     while stack and stack[-1][0] <= que[i]:
9         stack.pop()
10
11     if stack:
12         ff.append(stack[-1][1])
13     else:
```

```

14         ff.append(0)
15         stack.append((que[i],i+1))
16     ff.reverse()
17     for x in ff:
18         print(x,end = " ")
19

```

代码运行截图 (至少包含有"Accepted")

状态: Accepted

基本

源代码

```

N = int(input())
que = list(map(int,input().split()))

stack = []
ff = []
for i in range(N-1,-1,-1):
    while stack and stack[-1][0]<=que[i]:
        stack.pop()

    if stack:
        ff.append(stack[-1][1])
    else:
        ff.append(0)
    stack.append((que[i],i+1))
ff.reverse()
for x in ff:
    print(x,end = " ")

```

提

提

©2002-2022 POJ 京ICP备20010980号-1

09202: 舰队、海域出击!

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

思路:

代码

```

1  #
2  from collections import deque
3  T = int(input())
4  for i in range(T):
5      N,M = map(int,input().split())
6      graph = {j:[] for j in range(1,N+1)}
7      indegree = {j:0 for j in range(N+1)}
8      for j in range(M):
9          x,y = map(int,input().split())
10         graph[x].append(y)

```

```

11         indegree[y] += 1
12     que = deque([])
13     for j in range(1, N + 1):
14         if indegree[j] == 0:
15             que.append(j)
16     cnt = 0
17     while que:
18         s = que.popleft()
19         cnt += 1
20         for x in graph[s]:
21             indegree[x] -= 1
22             if indegree[x] == 0:
23                 que.append(x)
24     if cnt == N:
25         print("No")
26     else:
27         print("Yes")
28
29

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```

from collections import deque
T = int(input())
for i in range(T):
    N, M = map(int, input().split())
    graph = {j: [] for j in range(1, N + 1)}
    indegree = {j: 0 for j in range(N + 1)}
    for j in range(M):
        x, y = map(int, input().split())
        graph[x].append(y)
        indegree[y] += 1
    que = deque([])
    for j in range(1, N + 1):
        if indegree[j] == 0:
            que.append(j)
    cnt = 0
    while que:
        s = que.popleft()
        cnt += 1
        for x in graph[s]:
            indegree[x] -= 1
            if indegree[x] == 0:
                que.append(x)

```

04135: 月度开销

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

思路:

代码

```
1  #
2  N,M = map(int,input().split())
3  cost = [int(input()) for i in range(N)]
4  def day(maxn):
5      cnt = 1
6      temp = 0
7      for i in range(N):
8          if temp+cost[i]>maxn:
9              cnt+=1
10             temp = cost[i]
11         else:
12             temp+=cost[i]
13     return cnt
14
15  l = max(cost)
16  h = sum(cost)
17
18  while l<h:
19     m = (l+h)//2
20     if day(m)<=M:
21         h = m
22     else:
23         l=m+1
24  print(l)
25
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```
N,M = map(int,input().split())
cost = [int(input()) for i in range(N)]
def day(maxn):
    cnt = 1
    temp = 0
    for i in range(N):
        if temp+cost[i]>maxn:
            cnt+=1
            temp = cost[i]
        else:
            temp+=cost[i]
    return cnt

l = max(cost)
h = sum(cost)

while l<h:
    m = (l+h)//2
    if day(m)<=M:
        h = m
```

07735: 道路

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

思路:

代码

```
1 #
2 from heapq import *
3 #其实就是对堆写法的dijkstra的改写
4 K = int(input())
5 N = int(input())
6 R = int(input())
7 graph = [[] for i in range(N)]
8 for i in range(R):
9     S,D,L,T=map(int,input().split())
10    graph[S-1].append((D,L,T))
11    que = [(0,0,1)]
12
13
14    visited = [1000 for i in range(N)]
15    while que:
16        l,m,s = heappop(que)
17        if s ==N:
18            print(l)
```

```

19     exit()
20     if m>visited[s-1]:
21         continue
22     visited[s-1]=m
23
24     for newset,len,cost in graph[s-1]:
25         if m+cost<=K:
26             heappush(que,(l+len,m+cost,newset))
27
28     print(-1)

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

状态: Accepted

源代码

```

from heapq import *
#其实就是对堆写法的dijkstra的改写
K = int(input())
N = int(input())
R = int(input())
graph = [[] for i in range(N)]
for i in range(R):
    S,D,L,T=map(int,input().split())
    graph[S-1].append((D,L,T))
que = [(0,0,1)]

visited = [1000 for i in range(N)]
while que:
    l,m,s = heappop(que)
    if s ==N:
        print(l)
        exit()
    if m>visited[s-1]:
        continue
    visited[s-1]=m

    for newset,len,cost in graph[s-1]:
        if m+cost<=K:
            heappush(que,(l+len,m+cost,newset))

print(-1)

```


01182: 食物链

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

思路:

代码

```
1  #
2  N,K = map(int,input().split())
3  parent = [0]*(3*N+1)
4  for i in range(1,3*N+1):
5      parent[i]=i
6
7  def find(x):
8      if parent[x]!=x:
9          parent[x]=find(parent[x])
10     return parent[x]
11
12 def union(x,y):
13     parent[find(x)]=find(y)
14
15 cnt =0
16 for _ in range(K):
17     d,x,y = map(int,input().split())
18     if x > N or y > N:
19         cnt += 1
20         continue
21     if d == 1:
22
23         #如果有明显的信息说明x,y不是同类，那么为假话
24         if find(x)==find(y+N) or find(x)==find(y+2*N):
25             cnt+=1
26             continue
27         else:
28             union(x,y)
29             union(x+N,y+N)
30             union(x+2*N,y+2*N)
31     elif d == 2:
32         if find(x)==find(y) or find(x)==find(y+N):
33             cnt+=1
34             continue
35         else:
36             union(x,y+2*N)
37             union(x+N,y)
38             union(x+2*N,y+N)
39 print(cnt)
40
```

代码运行截图 (AC代码截图，至少包含有"Accepted")

状态: Accepted

基

源代码

```
N,K = map(int,input().split())
parent = [0]*(3*N+1)
for i in range(1,3*N+1):
    parent[i]=i

def find(x):
    if parent[x]!=x:
        parent[x]=find(parent[x])
    return parent[x]

def union(x,y):
    parent[find(x)]=find(y)

cnt =0
for _ in range(K):
    d,x,y = map(int,input().split())
    if x > N or y > N:
        cnt += 1
        continue
```

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。

上一次刚学的单调栈就用到了，自己写一遍以后原理就很清晰了