

Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Compiled by 赵策 数学科学学院

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

操作系统: Windows 11

Python编程环境: Visual Studio Code 1.86.2

1. 题目

22485: 升空的焰火，从侧面看

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

思路:

bfs, 存一下每层节点

代码

```
#
from collections import deque, defaultdict
def bfs(lst):
    queue=deque([(1,0)])
    res=defaultdict(list)
    while queue:
        node, level=queue.popleft()
        res[level].append(node)
        left, right=lst[node-1]
        if left!=-1:
            queue.append((left, level+1))
        if right!=-1:
            queue.append((right, level+1))
    return res
```

```
n=int(input())
lst=[map(int,input().split()) for _ in range(n)]
res=[d[-1] for d in bfs(lst).values()]
print(*res)
```

代码运行截图 (至少包含有"Accepted")

22485: 升空的焰火, 从侧面看

Accepted

28203: 【模板】单调栈

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

思路:

代码

```
#
from collections import deque
n=int(input())
res=[0]*n
nums=list(map(int,input().split()))
stack=[]
for i,t in enumerate(nums):
    while stack and t>nums[stack[-1]]:
        res[stack.pop()]=i+1
    stack.append(i)
print(*res)
```

代码运行截图 (至少包含有"Accepted")

28203: 【模板】单调栈

Accepted

09202: 舰队、海域出击!

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

思路: 下面给了两种dfs写法

最开始套无向图有环模板寄了一次, 刚刚改正确了, 如下:

```
#
def has_loop():
```

```

visited=[False]*n
def dfs(node,parent,visited_set):
    visited[node]=True
    visited_set.add(node)
    for neighbor in edges[node]:
        if not visited[neighbor]:
            if dfs(neighbor,node,visited_set):
                return True
        elif neighbor in visited_set:
            return True
    visited_set.remove(node)
    return False
for i in range(n):
    if not visited[i]:
        visited_set=set()
        if dfs(i,-1,visited_set):
            return 'Yes'
return 'No'

```

然后尝试top排序不知道为什么出问题了，最后还是优化初版的dfs，加了一个递归栈过了，晚点去学习一下top排序怎么写的（）

代码

```

#
def has_loop():
    visited=[False]*(n+1)
    recursion_stack=[False]*(n+1)
    def dfs(node):
        visited[node]=True
        recursion_stack[node]=True
        for nbr in edges[node]:
            if not visited[nbr]:
                if dfs(nbr):
                    return True
            elif recursion_stack[nbr]:
                return True
        recursion_stack[node]=False
        return False

    for node in range(1,n+1):
        if not visited[node]:
            if dfs(node):
                return 'Yes'
    return 'No'

for _ in range(int(input())):
    n,m=map(int,input().split())
    edges=[[] for _ in range(n+1)]
    for _ in range(m):
        x,y=map(int,input().split())
        edges[x].append(y)
    print(has_loop())

```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

09202: 舰队、海域出击!

Accepted

04135: 月度开销

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

思路:

binary_search, 这个check的作用就是检查在当前“最大值”下的最差划分能否继续分割, 即是否该划分不够m个fajo月, 不够说明当前“最大值”大了, 取right=mid继续二分, 反之取left=mid+1即可

```
#
def check(mid,m):
    total,count=0,0
    for expense in cost:
        total+=expense
        if total>mid:
            total=expense
            count+=1
    return count<m

def min_max_fajo():
    total=sum(cost)
    left,right=max(cost),total
    while left<right:
        mid=(left+right)//2
        if check(mid,m):
            right=mid
        else:
            left=mid+1
    return left

n,m=map(int,input().split())
cost=[int(input()) for _ in range(n)]
print(min_max_fajo())
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

04135: 月度开销

Accepted

07735: 道路

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

思路:

很明显的dijkstra

代码

```
#
import heapq
def dijkstra():
    heap=[(0,0,1)]
    visited=set()
    while heap:
        length, cost, city=heapq.heappop(heap)
        if city==n and cost<=k:
            return length
        if city in visited or cost>k:
            continue
        visited.add(city)
        for nbr, nlenth, ncost in edges[city]:
            if nbr not in visited and cost+ncost<=k:
                heapq.heappush(heap, (length+nlenth, cost+ncost, nbr))
        visited.remove(city)
    return -1

k=int(input())
n=int(input())
edges=[[[] for _ in range(n+1)]]
for _ in range(int(input())):
    s,d,l,t=map(int, input().split())
    edges[s].append((d,l,t))
print(dijkstra())
```

代码运行截图 (AC代码截图, 至少包含有"Accepted")

07735: 道路

Accepted

01182: 食物链

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

思路:

代码

```

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

n,k=map(int,input().split())
parent =list(range(3*n+1))
#分别用x+n和x+2n表示被x吃的和x吃的种类，很好的想法
f_count=0
for _ in range(k):
    d,x,y=map(int,input().split())
    if x>n or y>n:
        f_count+=1
        continue
    if d==1:
        if find(x+n)==find(y) or find(y+n)==find(x):
            f_count+=1
            continue
        #x和y为同类时合并x和y以及其吃与被吃的种类
        parent[find(x)]=find(y)
        parent[find(x+n)]=find(y+n)
        parent[find(x+2*n)]=find(y+2*n)
    else:
        if find(x)==find(y) or find(y+n)==find(x):
            f_count+=1
            continue
        parent[find(x)]=find(y+2*n)
        parent[find(x+n)]=find(y)
        parent[find(x+2*n)]=find(y+n)
print(f_count)

```

代码运行截图 (AC代码截图，至少包含有"Accepted")

01182: 食物链

Accepted

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。

```

# 月度开销有一种感觉可行的dp写法，但是由于数据过大导致二维数组MLE，最后只能改写二分查找了つ__c
def min_max_fajo():
    dp=[[float('inf')]*(m+1) for _ in range(n+1)]
    dp[0][0]=0
    for i in range(1,n+1):
        dp[i][1]=sum(cost[:i])
    for j in range(2,m+1):
        for i in range(j,n+1):
            for k in range(i-1,-1,-1):

```

```
        curr_sum=sum(cost[k:i])
        dp[i][j]=min(dp[i][j],max(dp[k][j-1],curr_sum))

    return dp[n][m]

n,m=map(int,input().split())
cost=[int(input()) for _ in range(n)]
print(min_max_fajo())
```