# 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：

代码

```
#
from collections import deque
class Treenode:
    def __init__(self,value):
        self.value=value
        self.left=None
        self.right=None
        self.height=2
        self.parent=None
n=int(input())
lst=[Treenode(i) for i in range(1,n+1)]
```

```
lst[0].height=1
for i in range(n):
    a,b=map(int,input().split())
    if a!=-1:
        lst[i].left=lst[a-1]
        lst[a-1].parent=lst[i]
    if b!=-1:
        lst[i].right=lst[b-1]
        lst[b-1].parent=lst[i]
for i in range(1,n):
    a=lst[i].parent
    while lst[i].parent!=lst[0]:
        lst[i].height+=1
        lst[i].parent=lst[i].parent.parent
    lst[i].parent=a
root=lst[0]
def bfs(root):
    q=deque()
    q.append(root)
    result=[]
    while q:
        a=q.popleft()
        if a.left!=None:
            q.append(a.left)
        if a.right!=None:
            q.append(a.right)
        if q:
            b=q.popleft()
            if a.height==b.height-1:
                result.append(str(a.value))
            q.appendleft(b)
        else:
            result.append(str(a.value))
            return result
result=bfs(root)
print(" ".join(result))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

基本信息

#: 45106887
题目: 22485
提交人: wangyecheng
内存: 3696kB
时间: 27ms
语言: Python3
提交时间: 2024-05-27 15:21:50

源代码

```
from collections import deque
class Treenode:
    def __init__(self,value):
        self.value=value
        self.left=None
        self.right=None
        self.height=2
        self.parent=None
n=int(input())
lst=[Treenode(i) for i in range(1,n+1)]
lst[0].height=1
for i in range(n):
    a,b=map(int,input().split())
    if a!=-1:
        lst[i].left=lst[a-1]
        lst[a-1].parent=lst[i]
    if b!=-1:
        lst[i].right=lst[b-1]
        lst[b-1].parent=lst[i]
for i in range(1,n):
    a=lst[i].parent
    while lst[i].parent!=lst[0]:
        lst[i].height+=1
        lst[i].parent=lst[i].parent.parent
    lst[i].parent=a
root=lst[0]
def bfs(root):
    q=deque()
    q.append(root)
    result=[]
    while q:
        a=q.popleft()
        if a.left!=None:
            q.append(a.left)
        if a.right!=None:
            q.append(a.right)
        if q:
            b=q.popleft()
            if a.height==b.height-1:
                result.append(str(a.value))
            q.appendleft(b)
        else:
            result.append(str(a.value))
```

# 28203:【模板】单调栈

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

思路:

代码

```
#
n=int(input())
lst=list(map(int,input().split()))
if n==1:
    print(0)
else:
    result=["0"]*n
    stack=[]
    for i in range(n):
        cur=lst[i]
        while stack and cur>lst[stack[-1]]:
            index=stack.pop()
```

```
            result[index]=str(i+1)
        stack.append(i)
    print(" ".join(result))
```

代码运行截图 ==（至少包含有"Accepted"）==

状态: Accepted

源代码

```
n=int(input())
lst=list(map(int,input().split()))
if n==1:
    print(0)
else:
    result=["0"]*n
    stack=[]
    for i in range(n):
        cur=lst[i]
        while stack and cur>lst[stack[-1]]:
            index=stack.pop()
            result[index]=str(i+1)
        stack.append(i)
    print(*result)
```

基本信息

| | |
|---|---|
| #: | 45107081 |
| 题目: | 28203 |
| 提交人: | wangyecheng |
| 内存: | 397312kB |
| 时间: | 3420ms |
| 语言: | Python3 |
| 提交时间: | 2024-05-27 15:47:38 |

## 09202: 舰队、海域出击!

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

思路:

代码

```
#
from collections import defaultdict

def dfs(node, color):
    color[node] = 1
    for neighbour in graph[node]:
        if color[neighbour] == 1:
            return True
        if color[neighbour] == 0 and dfs(neighbour, color):
            return True
    color[node] = 2
    return False

T = int(input())
for _ in range(T):
    N, M = map(int, input().split())
    graph = defaultdict(list)
    for _ in range(M):
        x, y = map(int, input().split())
        graph[x].append(y)
    color = [0] * (N + 1)
    is_cyclic = False
    for node in range(1, N + 1):
```

```
        if color[node] == 0:
            if dfs(node, color):
                is_cyclic = True
                break
    print("Yes" if is_cyclic else "No")
```

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 defaultdict

def dfs(node, color):
    color[node] = 1
    for neighbour in graph[node]:
        if color[neighbour] == 1:
            return True
        if color[neighbour] == 0 and dfs(neighbour, color):
            return True
    color[node] = 2
    return False

T = int(input())
for _ in range(T):
    N, M = map(int, input().split())
    graph = defaultdict(list)
    for _ in range(M):
        x, y = map(int, input().split())
        graph[x].append(y)
    color = [0] * (N + 1)
    is_cyclic = False
    for node in range(1, N + 1):
        if color[node] == 0:
            if dfs(node, color):
                is_cyclic = True
                break
    print("Yes" if is_cyclic else "No")
```

基本信息

| | |
|---|---|
| #: | 45109421 |
| 题目: | 09202 |
| 提交人: | wangyecheng |
| 内存: | 53376kB |
| 时间: | 3718ms |
| 语言: | Python3 |
| 提交时间: | 2024-05-27 19:40:09 |

## 04135: 月度开销

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

思路:

代码

```
#
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]
lo = max(expenditure)
hi = sum(expenditure) + 1
ans = 1
while lo < hi:
    mid = (lo + hi) // 2
    if check(mid):
        lo = mid + 1
    else:
        ans = mid
        hi = mid

print(ans)
```

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())
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]
lo = max(expenditure)
hi = sum(expenditure) + 1
ans = 1
while lo < hi:
    mid = (lo + hi) // 2
    if check(mid):
        lo = mid + 1
    else:
        ans = mid
        hi = mid

print(ans)
```

基本信息

| | |
|---|---|
| #: | 45109669 |
| 题目: | 04135 |
| 提交人: | wangyecheng |
| 内存: | 7484kB |
| 时间: | 515ms |
| 语言: | Python3 |
| 提交时间: | 2024-05-27 19:56:37 |

## 07735: 道路

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

思路:

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"）==

状态: Accepted

源代码

```
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)
```

基本信息

| | |
|---|---|
| #: | 45109681 |
| 题目: | 07735 |
| 提交人: | wangyecheng |
| 内存: | 6536kB |
| 时间: | 45ms |
| 语言: | Python3 |
| 提交时间: | 2024-05-27 19:57:43 |

# 01182: 食物链

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

思路:

代码

```
#
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"）==

状态：Accepted

源代码

```python
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)
```

基本信息

| | |
|---|---|
| #: | 45109700 |
| 题目: | 01182 |
| 提交人: | wangyecheng |
| 内存: | 9376kB |
| 时间: | 509ms |
| 语言: | Python3 |
| 提交时间: | 2024-05-27 19:59:15 |

# 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。==

本周事情比较多，后面四题都是参考题解的，马上机考了，还要更加努力