# Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied 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/

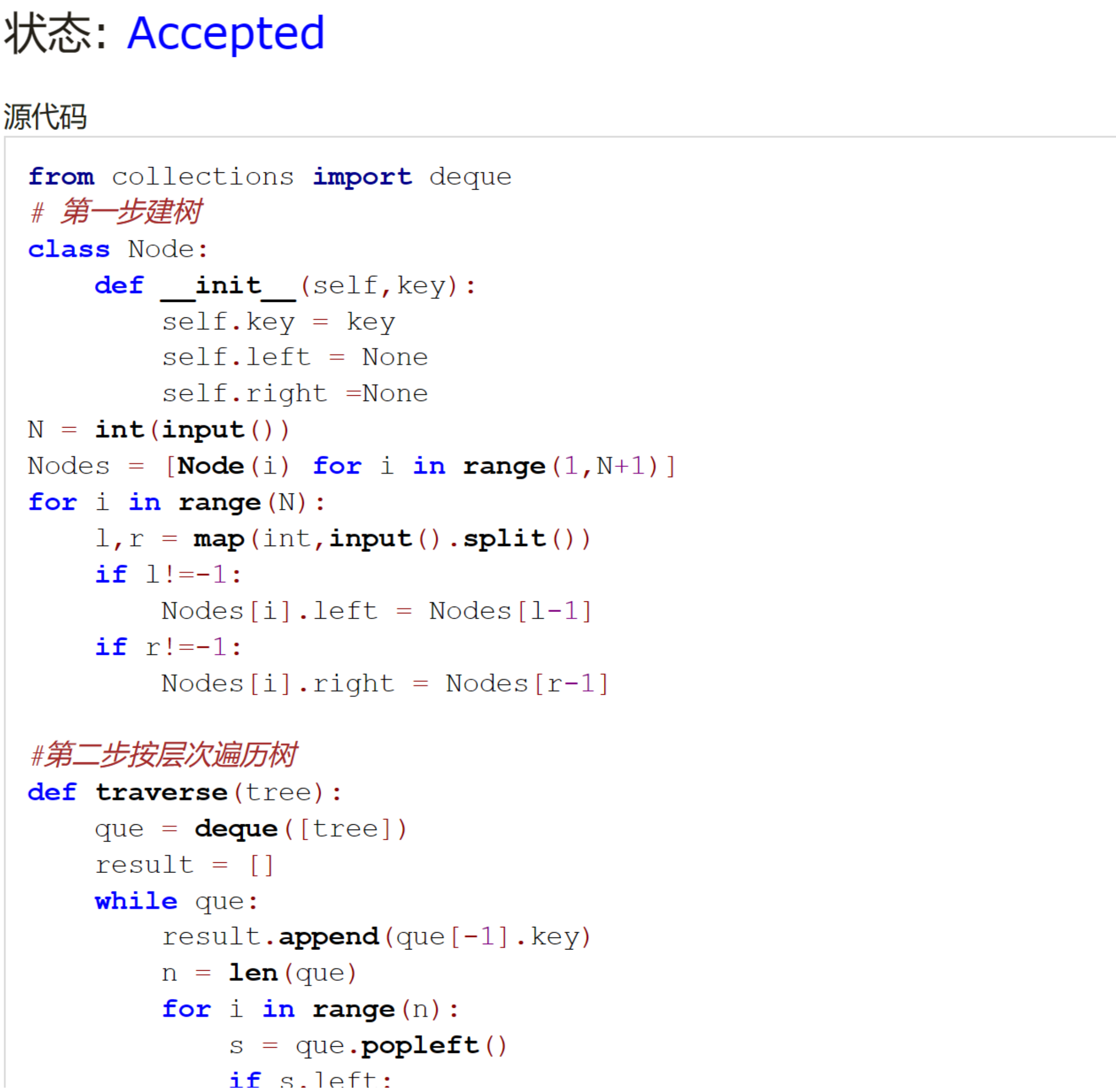
思路：

代码

```python
#
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())
```

```python
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:
                que.append(s.left)
            if s.right:
                que.append(s.right)
    for x in result:
        print(x,end = " ")
traverse(Nodes[0])


```

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/

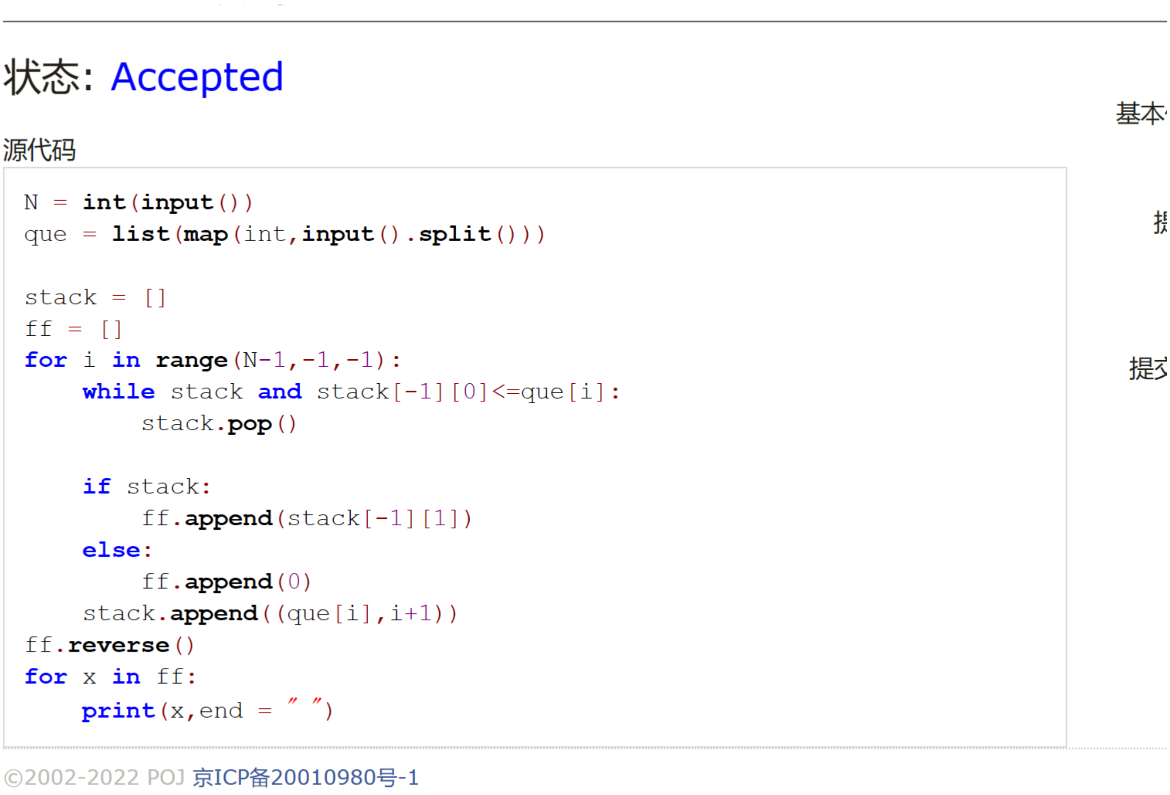
思路:

代码

```
#
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:
```

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

```

代码运行截图 （至少包含有"Accepted"）



状态: Accepted

源代码

```python
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 = ” ”)
```

©2002-2022 POJ 京ICP备20010980号-1

## 09202: 舰队、海域出击！

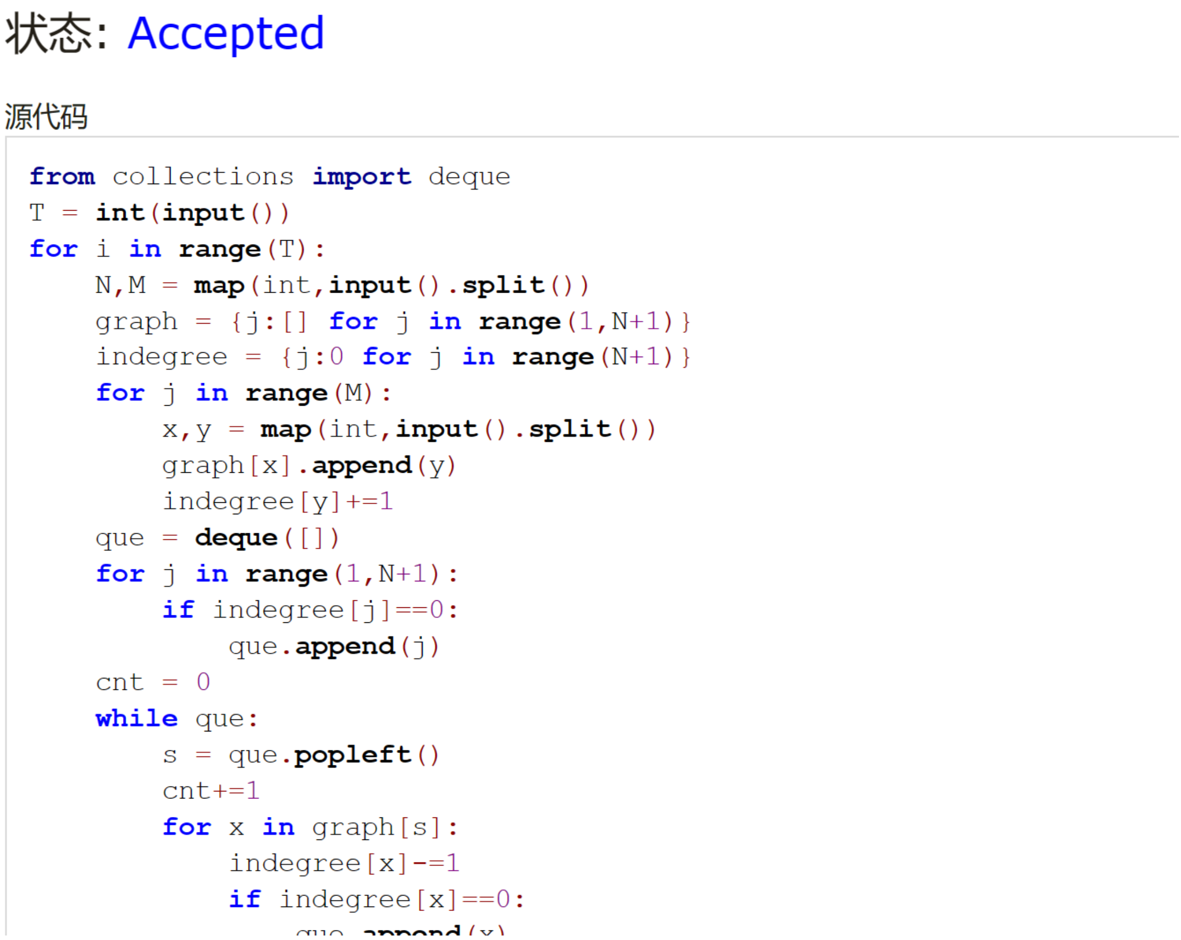
http://cs101.openjudge.cn/practice/09202/

思路：

代码

```python
#
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)
    if cnt == N:
        print("No")
    else:
        print("Yes")


```

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

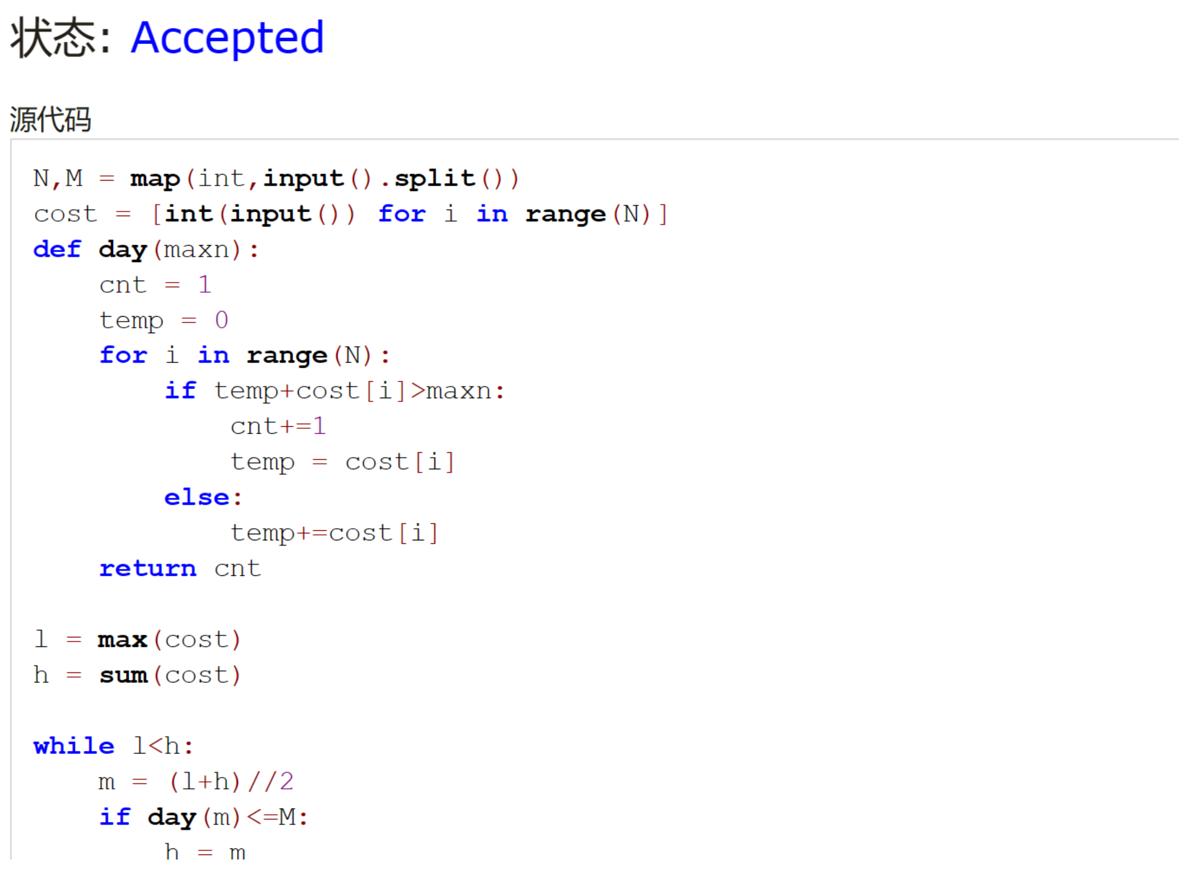
http://cs101.openjudge.cn/practice/04135/

思路：

代码

```python
#
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
    else:
        l=m+1
print(l)

```

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/

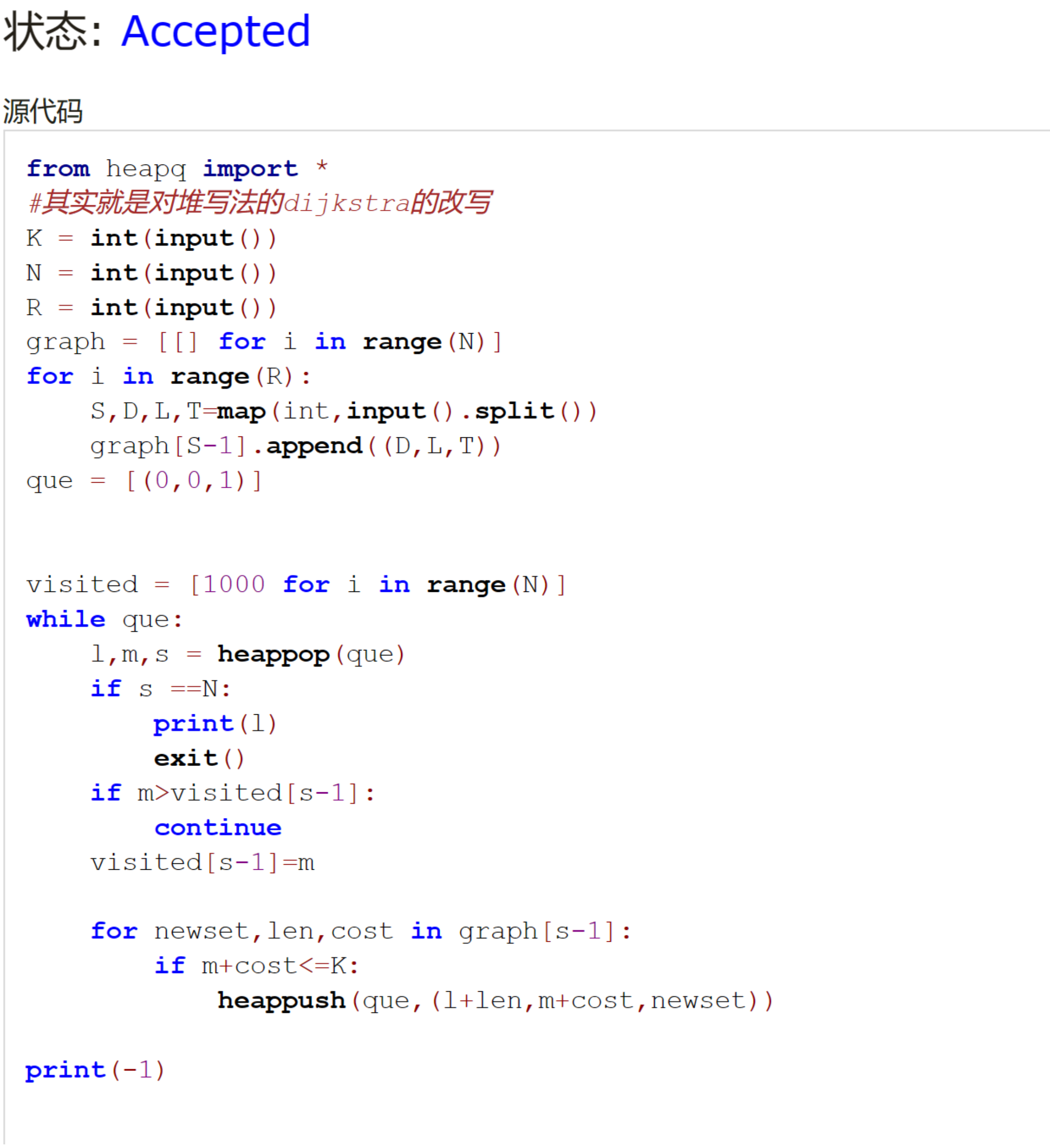
思路：

代码

```
#
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)
```

```python
        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)
```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

源代码

```python
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

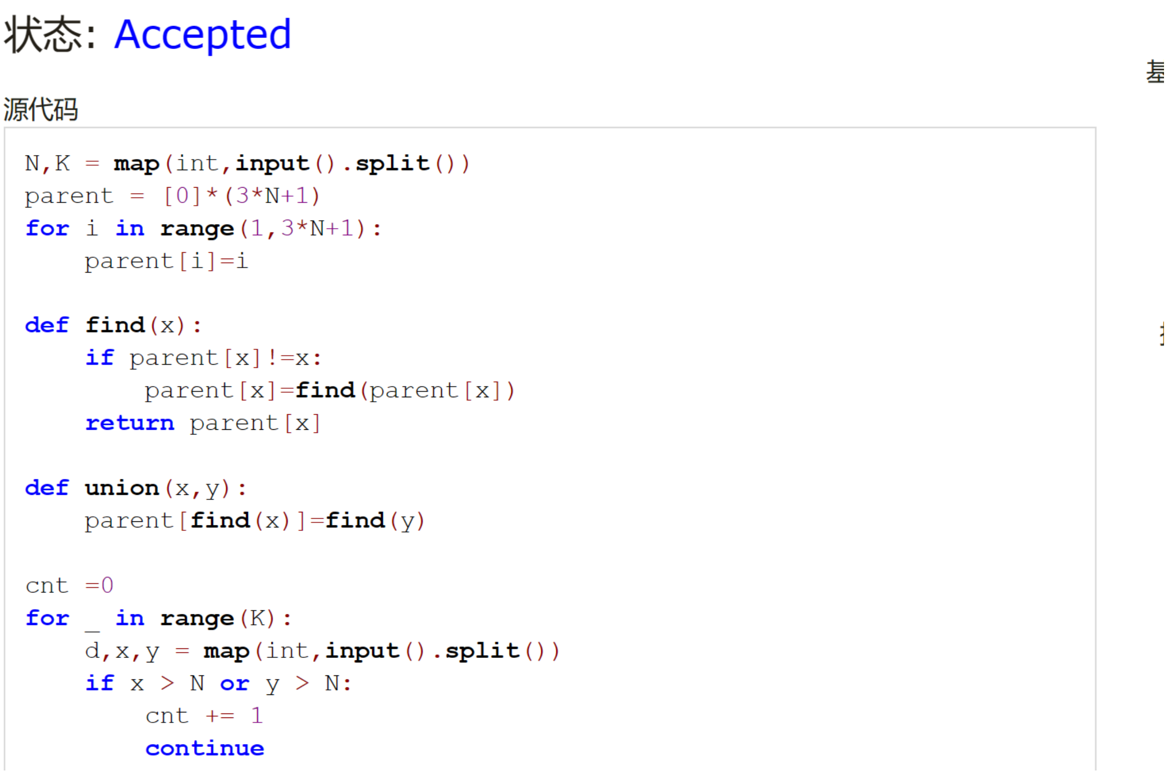
http://cs101.openjudge.cn/practice/01182/

思路：

代码

```python
#
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
    if d == 1:

        #如果有明显的信息说明x,y不是同类，那么为假话
        if find(x)==find(y+N) or find(x)==find(y+2*N):
            cnt+=1
            continue
        else:
            union(x,y)
            union(x+N,y+N)
            union(x+2*N,y+2*N)
    elif d == 2:
        if find(x)==find(y) or find(x)==find(y+N):
            cnt+=1
            continue
        else:
            union(x,y+2*N)
            union(x+N,y)
            union(x+2*N,y+N)
print(cnt)

```

代码运行截图 （AC代码截图，至少包含有"Accepted"）



状态: Accepted

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