Assignment #F: All-Killed 满分

Updated 1844 GMT+8 May 20, 2024

2024 spring, Complied by 余汶青 2300012265

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：版本　Windows 11 家庭中文版
版本　22H2
安装日期 2023/7/18
操作系统版本 22621.2283
序列号　5CD323PJKL
体验　Windows Feature Experience Pack 1000.22662.1000.0

Python编程环境：* Spyder version: 5.4.3 (conda)

- Python version: 3.11.4 64-bit
- Qt version: 5.15.2
- PyQt5 version: 5.15.7
- Operating System: Windows 10

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,key):
        self.key=key
        self.left=None
```

```
        self.right=None
        self.father=None

n=int(input())
tree={}
for i in range(1,1+n):
    tree[i]=treenode(i)
for i in range(1,1+n):
    a,b=map(int,input().split())
    if a!=-1:
        tree[i].left=tree[a]
    if b!=-1:
        tree[i].right=tree[b]

q=deque()
q.append(tree[1])
print(1,end=' ')
while q:
    a=[]
    for i in range(len(q)):
        now=q.popleft()
        if now.left!=None:
            a.append(now.left)
        if now.right!=None:
            a.append(now.right)
    if a:
        print(a[-1].key,end=' ')
    q.extend(a)
```

代码运行截图

#45076095提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
from collections import deque
class treenode:
    def __init__(self,key):
        self.key=key
        self.left=None
        self.right=None
        self.father=None
```

基本信息

#: 45076095
题目: 22485
提交人: 23n2300012265
内存: 3784kB
时间: 21ms
语言: Python3
提交时间: 2024-05-25 11:05:47

28203:【模板】单调栈

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

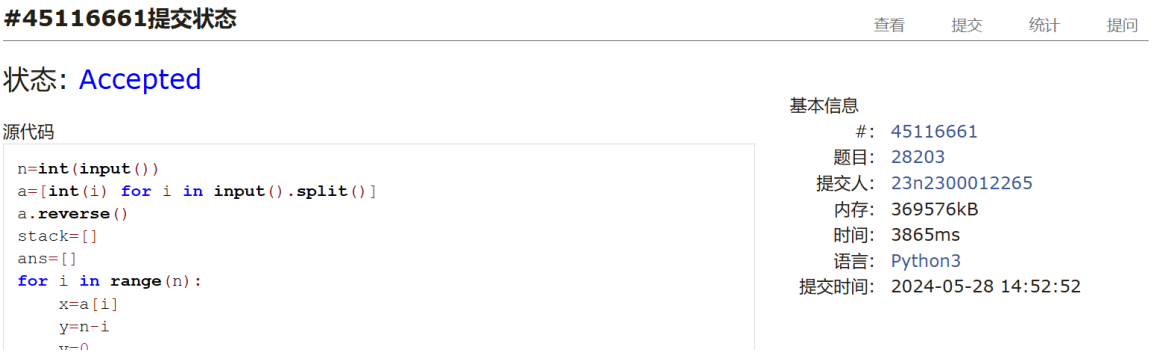
思路:

代码

```
n=int(input())
a=[int(i) for i in input().split()]
```

```
a.reverse()
stack=[]
ans=[]
for i in range(n):
    x=a[i]
    y=n-i
    v=0
    while stack:
        nx,ny=stack.pop()
        if nx>x:
            v=ny
            stack.append((nx,ny))
            break
    ans.append(v)
    stack.append((x,y))
ans.reverse()
print(*ans,sep=' ')
```

代码运行截图



#45116661提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
n=int(input())
a=[int(i) for i in input().split()]
a.reverse()
stack=[]
ans=[]
for i in range(n):
    x=a[i]
    y=n-i
    v=0
```

基本信息

#: 45116661
题目: 28203
提交人: 23n2300012265
内存: 369576kB
时间: 3865ms
语言: Python3
提交时间: 2024-05-28 14:52:52

09202: 舰队、海域出击!

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

思路:

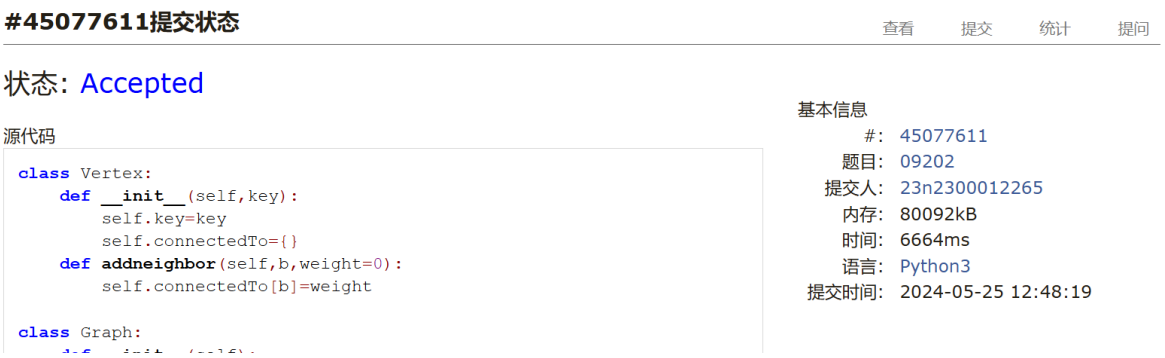
注意这是有向图，需要每次bfs的时候单独标一下这次bfs经过的点

代码

```
class Vertex:
    def __init__(self,key):
        self.key=key
        self.connectedTo={}
    def addneighbor(self,b,weight=0):
        self.connectedTo[b]=weight

class Graph:
    def __init__(self):
        self.vertList={}
```

```
        self.numVertices=0
    def addVertex(self,key):
        newvertex=Vertex(key)
        self.vertList[key]=newvertex
        self.numVertices+=1
    def addEdge(self,a,b,weight=0):
        if a not in self.vertList:
            self.addVertex(a)
        if b not in self.vertList:
            self.addVertex(b)
        self.vertList[a].addneighbor(self.vertList[b],weight)

ans=1
visited=[]

def dfs(a,nn):
    global ans,visited
    if a in route.vertList:
        visited[a]=2
        #visi=[0 for i in range(nn+1)]
        #visi[a]=1
        nexta=route.vertList[a].connectedTo
        for i in nexta.keys():
            if visited[i.key]==2:
                ans=0
                return 1
            if visited[i.key]==1:
                continue

            else:
                visited[i.key]=2
                if dfs(i.key,nn):
                    return 1
        visited[a]=1


from collections import deque
T=int(input())
for _ in range(T):
    ans=1
    n,m=map(int,input().split())
    route=Graph()
    for i in range(m):
        a,b=map(int,input().split())
        route.addEdge(a,b)
    q=deque()
    visited=[0 for i in range(n+1)]
    for i in range(1,n+1):
        if visited[i]==0:
            dfs(i,n)
    if ans==0:
        print("Yes")
    else:
        print("No")
```

代码运行截图



04135: 月度开销

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

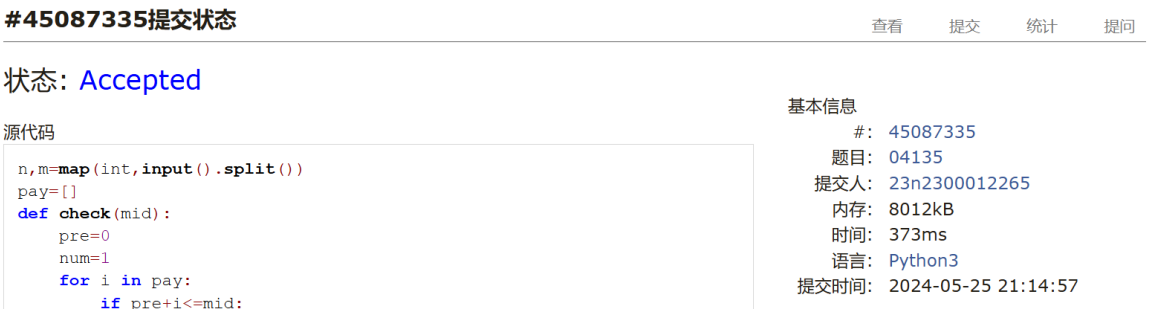
思路:

尝试用最小堆来写尝试了近三个小时，一直WA后惊觉最小堆没法解决两种连接方式开销相同的情况，看题解恍然大悟原来要用二分查找，然后十分钟写完。。。（选择大于努力。。。。。。。。。。。。）

代码

```
n,m=map(int,input().split())
pay=[]
def check(mid):
    pre=0
    num=1
    for i in pay:
        if pre+i<=mid:
            pre+=i
        else:
            num+=1
            pre=i
    if num<=m:
        return 1
    else:
        return 0

for i in range(n):
    pay.append(int(input()))
right=sum(pay)
left=max(pay)
while left<right:
    mid=(right+left)//2
    #print(mid)
    if check(mid):
        right=mid
    else:
        left=mid+1
print(right)
```

代码运行截图



07735: 道路

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

思路:

最开始觉得这个应该是Dijkstra的变体，但又觉得可以当成背包问题，于是先用dp做了半天，然后一直超时，就卡在1000多一点，但是怎么都没法降下来，折腾了两个小时后换成Dijkstra

反思：看见容量超过1000就不要再想着背包了

代码

```
import heapq

class vertex():
    def __init__(self,k):
        self.edge=[]#前一个点编号：（花费，路程）
        self.key=k
        self.dis=-1
        self.cost=-1

K=int(input())
N=int(input())
R=int(input())
graph={}
for i in range(1,N+1):
    graph[i]=vertex(i)
for i in range(R):
    s,d,l,t=map(int,input().split())
    graph[s].edge.append((d,l,t))
graph[1].dis=0
graph[1].cost=0
p=[]
heapq.heappush(p,(0,1,0))
```

```
while p:
    nowdis,nowver,nowcost=heapq.heappop(p)
    if nowver==N:
        graph[N].dis=nowdis
        break
    for k,l,t in graph[nowver].edge:
            if nowcost+t<=K:
                #if nowdis+l<graph[k].dis:
                    graph[k].dis=nowdis+l
                    graph[k].cost=nowcost+t
                    heapq.heappush(p,(graph[k].dis,k,graph[k].cost))
print(graph[N].dis)
```

代码运行截图

01182: 食物链

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

思路:

思路不难想，就是给并查集多增一个eat和eaten，但是具体实现需要考虑各种情况。第一版代码考虑的是什么情况下是真话，花了二十分钟写完，然后调试了一个小时发现情况不好分类。于是改变思路考虑什么时候是假话，又调试了一个小时，一直WA，要到测试数据一组组测发现只有第四组错了，说明我的大体思路是正确的。然后发现好像是某几处少了一些判断，但这些判断和前面重复了，于是又花了五分钟改成函数/递归写法，终于AC了，从想思路到AC前前后后可能花了一共三多个小时，但收获很大。（不过感觉现在这方法挺笨拙的。）

小插曲：一直WA又没有测试数据的时候想着上洛谷看看有没有这道题（可以下测试数据），然后发现我居然初三的时候AC过，不知道当时怎么做出来的。。。

代码

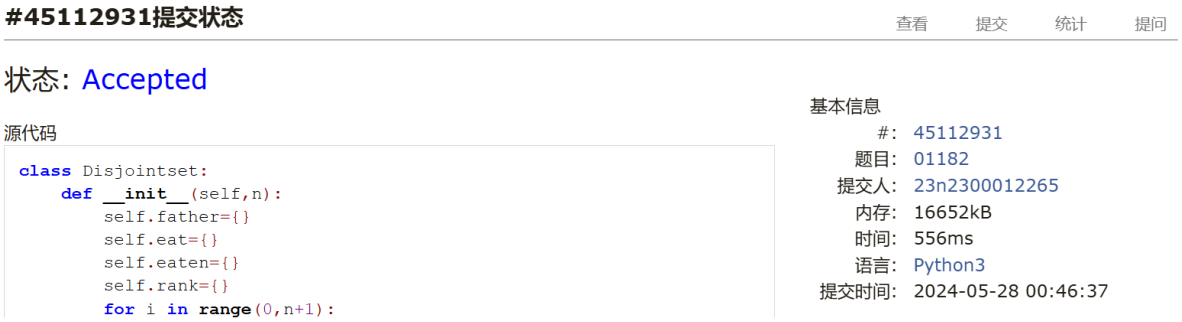
```
class Disjointset:
    def __init__(self,n):
        self.father={}
        self.eat={}
        self.eaten={}
        self.rank={}
```

```
        for i in range(0,n+1):
            self.father[i]=i
            self.eat[i]=0
            self.eaten[i]=0
            self.rank[i]=0
    def findfather(self,i):
        if self.father[i]!=i:
            self.father[i]=self.findfather(self.father[i])
            self.eat[i]=self.eat[self.father[i]]
            self.eaten[i]=self.eaten[self.father[i]]
        return self.father[i]
    def union(self,a,b,k=1):
        fa=self.findfather(a)
        fb=self.findfather(b)
        featfa=self.findfather(self.eat[fa])
        featfb=self.findfather(self.eat[fb])
        featenfa=self.findfather(self.eaten[fa])
        featenfb=self.findfather(self.eaten[fb])
        if fa==fb:
            return
        v1=0
        if featfa==0 and featfb!=0:
            self.eat[fa]=featfb
        elif featfb==0 and featfa!=0:
            self.eat[fb]=featfa
        elif featfa!=0 and featfb!=0:
            v1=1
        v2=0
        if featenfa==0 and featenfb!=0:
            self.eaten[fa]=featenfb
        elif featenfb==0 and featenfa!=0:
            self.eaten[fb]=featenfa
        elif featenfa!=0 and featenfb!=0:
            v2=1
        if self.rank[fa]>self.rank[fb]:
            self.father[fb]=fa
        elif self.rank[fa]<self.rank[fb]:
            self.father[fa]=fb
        else:
            self.father[fb]=fa
            self.rank[fa]+=1
        if v1:
            self.union(featfa,featfb,2)
        if v2:
            self.union(featenfa,featenfb,2)

N,K=map(int,input().split())
foodchain=Disjointset(N)
lie=0

def check1(x,y):
    global lie,foodchain
    fx=foodchain.findfather(x)
    fy=foodchain.findfather(y)
    if fx!=fy:
        eatfx=foodchain.findfather(foodchain.eat[fx])
```

```
        eatfy=foodchain.findfather(foodchain.eat[fy])
        eatenfx=foodchain.findfather(foodchain.eaten[fx])
        eatenfy=foodchain.findfather(foodchain.eaten[fy])
        if eatfx==fy or eatfy==fx:#不能互相吃
            lie+=1
            #print(2,lie)
            return
        elif (eatfx==eatenfy and eatfx!=0) or (eatfy==eatenfx and eatfy!=0):#不能
成链
            lie+=1
            #print(3,lie)
            return
        foodchain.union(fx,fy)
def check2(x,y):
    global lie,foodchain
    fx=foodchain.findfather(x)
    fy=foodchain.findfather(y)
    eatfx=foodchain.findfather(foodchain.eat[fx])
    eatfy=foodchain.findfather(foodchain.eat[fy])
    eatenfx=foodchain.findfather(foodchain.eaten[fx])
    eatenfy=foodchain.findfather(foodchain.eaten[fy])
    if eatfx!=fy:
        if eatfy==fx:#不能反过来吃
            lie+=1
            #print(4,lie)
            return
        elif fx==fy:#不能是同类
            lie+=1
            #print(5,lie)
            return
        elif eatfx==eatenfy and eatfx!=0:#不能成链
            lie+=1
            #print(6,lie)
            return
        else:
            foodchain.eat[fx]=fy
            foodchain.eaten[fy]=fx
            if eatenfy!=0:
                foodchain.union(fx,eatenfy)
            if eatfx!=0:
                foodchain.union(fy,eatfx)
            if eatenfx!=0:
                if eatfy!=0:
                    check1(eatenfx,eatfy)
                else:
                    check2(fy,eatenfx)
            if eatfy!=0:
                if eatenfx!=0:
                    check1(eatenfx,eatfy)
                else:
                    check2(eatfy,fx)

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

```
            #print(1,lie)
            continue
        #print(foodchain.father)
        #print(foodchain.eat)
        #print(foodchain.eaten)
        if d==1:
            check1(x,y)
        if d==2:
            check2(x,y)

print(lie)
```

代码运行截图



#45112931提交状态

查看 提交 统计 提问

状态: Accepted

源代码

```
class Disjointset:
    def __init__(self,n):
        self.father={}
        self.eat={}
        self.eaten={}
        self.rank={}
        for i in range(0,n+1):
```

基本信息

#:	45112931
题目:	01182
提交人:	23n2300012265
内存:	16652kB
时间:	556ms
语言:	Python3
提交时间:	2024-05-28 00:46:37

2. 学习总结和收获

不简单，写吐了