机考小遗憾

28013:堆路径OpenJudge - 28013:堆路径

思路：dfs，从右往左遍历，注意深浅拷贝问题，sort与sorted的区别

代码：

```
from collections import deque
n=int(input())
sample=[int(x) for x in input().split()]
class tree:
    def __init__(self,name):
        self.name=name
        self.left=None
        self.right=None
def buildtree(sample):
    sample1=[tree(i) for i in sample]
    queue=deque()
    queue.append(sample1.pop(0))
    finished=[]
    while queue:
        a=queue.popleft()
        if not a.left and sample1:
            tempo=sample1.pop(0)
            queue.append(tempo)
            a.left=tempo
        if not a.right and sample1:
            tempo=sample1.pop(0)
            queue.append(tempo)
            a.right=tempo
        if a.right and a.left:
            finished.append(a)
    return finished[0]
ans=[]
def dfs(root,path):
    global ans
    if not root.left and not root.right:

        ans.append(path.copy())
        return
    if root.right:
        path.append(root.right.name)
        dfs(root.right,path)
        path.pop()
    if root.left:
        path.append(root.left.name)
        dfs(root.left,path)
        path.pop()

dfs(buildtree(sample),[sample[0]])

flag1=1
flag2=2
for i in ans:
```

```
    new=sorted(i)#判断小根堆
    new1=sorted(i,reverse=True)
    print(*i)
    if flag1==1:
        if new!=i:
            flag1=0

    if flag2==2:
        if new1!=i:
            flag2=0
if flag1==1:
    print("Min Heap")
elif flag2==2:
    print("Max Heap")
else:
    print("Not Heap")
```

截图：

28193:谣言[OpenJudge - 28193:谣言]

思路：并查集，然后赋最小值

代码：

```
import heapq
class unionandfind:
    def __init__(self,n):
        self.fathers=[i for i in range(n)]
    def find(self,a):
        if self.fathers[a]!=a:
            self.fathers[a]=self.find(self.fathers[a])
        return self.fathers[a]
```

```
    def union(self,a,b):
        a_fa=self.find(a)
        b_fa=self.find(b)
        if a_fa!= b_fa:
            self.fathers[a_fa]=b_fa

n,m=map(int,input().split())
money=[int(x) for x in input().split()]
uf=unionandfind(n)
for i in range(m):
    xi,yi=map(int,input().split())
    xi,yi=xi-1,yi-1
    uf.union(xi,yi)

a=[uf.find(x) for x in range(n)]
needmoney={}
idx={}
newa=set(a)
for i in newa:
    needmoney[i]=[]#记录每一个连通分量需要的钱数列表
for m in range(n):
    needmoney[a[m]].append(money[m])
ans=[]
for m in needmoney.keys():
    ans.append(min(needmoney[m]))

print(sum(ans))
```

截图: