Wangjihe的互测

|  |  |  |  |
| --- | --- | --- | --- |
| 题目名称 | install | mitm | dictionary |
| 输入文件 | install.in | mitm.in | dictionary.in |
| 输出文件 | install.out | mitm.out | dictionary.out |
| 每个测试点时限 | 1s | 1s | 1s |
| 测试点数目 | 20 | 20 | 10 |
| 每个测试点分值 | 5 | 5 | 10 |
| 运行内存上限 | 256M | 256M | 256M |

**注意:难度与题目顺序无关,请自行判断.**

评测在Windows10 x64 lemon下测试.

不需要建立子文件夹.

编译命令: **g++ install.cpp -o install -O2**

install

**Description**

wangjihe正策划着一场攻击,目标是夺取xzms网络的最高权限.

他为了收集情报,假装帮老师安装软件.虽然wangjihe很想偷完资料就跑,但为了防止被怀疑,他还是不得不完成这个任务.

现在我们的手头有N个软件,对于一个软件i,它要占用Wi的磁盘空间,它的价值为Vi.我们希望从中选择一些软件安装到一台磁盘容量为M计算机上,使得这些软件的价值和尽可能大（即Vi的和最大）.

但是现在有个问题：软件之间存在依赖关系,即软件i只有在安装了软件j（包括软件j的直接或间接依赖）的情况下才能正确工作.幸运的是,一个软件最多依赖另外一个软件(可能有环).如果一个软件不能正常工作,那么它能够发挥的作用为0.

我们现在知道了软件之间的依赖关系：软件i依赖软件Di.现在请你设计出一种方案,安装价值尽量大的软件.一个软件最多只能被安装一次,如果一个软件没有依赖则Di=0,这时只要这个软件安装了,它就能正常工作.

**Input**

第一行包含两个整数N, M

第二,三,四行均包含N个整数,分别代表W,V,D

**Output**

一个整数,表示最大价值.

**Sample Input**

3 10

5 5 6

2 3 4

0 1 1

**Sample Output**

5

**Hint**

对于10%的数据, N<=18

对于另外20%的数据, 保证没有环

对于100%的数据, 0<=N<=100, 0<=M<=500, 0<=Wi<=M, 0<=Vi<=1000, 0<=Di<=N, Di≠i

**mitm**

**Description**

wangjihe通过前期的情报准备工作,调查好了整个xzms的网络结构.

整个网络共有N台终端,由M条网线连接,其中有一些网线是10MBPs的低速网线,剩下的则是100MBPs的高速网线.

由于wangjihe能力不足,他决定仅保留必要的N-1条网线,对它们进行控制,使N台终端保持连接,把剩下的直接网线切断.

为避免网络过卡导致刷卡机故障,被学生和老师发现,经过详细的计算,wangjihe得出剩下的网线组成的网络中,恰好有K条10MBPs的网线最优,

他希望求出**任意一种方案**能满足以上条件,来评估这次攻击的可行性**,如果没有任何方案,请只输出-1**.

为了在被发现时分担尽量少的责任,他把这个任务交给了你.

**Input**

第一行包含三个正整数N,M和K.

接下来的M行,每行有三个数,Ai,Bi,Ti,若Ti=0,表示Ai与Bi之间有一条10MBPs的网线, 若Ti=1,表示Ai与Bi之间有一条100MBPs的网线.**输入保证两台终端最多只有一条直接相连的网线.**

**Output**

共N-1行,每行三个数,格式与输入相同,表示一种可行方案.

**Sample Input**

5 7 2

1 3 0

4 5 1

3 2 0

5 3 1

4 3 0

1 2 1

4 2 1

**Sample Output**

3 2 0

4 3 0

5 3 1

1 2 1

**Hint**

**本题存在spj**

对于30%的数据,N<=10.

对于50%的数据,N<=1000

对于100%的数据,1<=N<=20000,1<=M<= 100000,0<=K<=N-1

dictionary

**Description**

wangjihe成功发动中间人攻击后,决定拿它们去撞库.于是他用爬虫在网上广泛收集用户名,制成了一个用户名字典,希望能在其中找到相似的账户.

由于他正沉迷文明五,就把这个任务丢给了你.具体是这样的,输入N个找到的用户名,作为字典,然后再输入M个字符串,分别在字典中查询与这M个字符串编辑距离为1的用户名个数,如果它是字典中的单词,则输出-1.

在这里,字符串a与b的编辑距离是指:可以通过以下操作,将a变为b或b变为a,最少操作次数即为编辑距离距离.

1. 删除a串中某个位置的字母;
2. 添加一个字母到a串中某个位置;
3. 替换a串中某一位置的一个字母为另一个字母;

**Input**

第一行包含两个正整数N和M.

接下来的N行,每行一个字符串,表示一个字典中的用户名.

再接下来M行,每行一个字符串,表示一个待查字符串.

**Output**

输出应包括M行,表示每个询问的答案.

**Sample Input**

4 3

abcd

abcde

aabc

abced

abcd

abc

abcdd

**Sample Output**

-1

2

3

**Hint**

对于50%的数据,0 <= N,M <= 1000

对于100%的数据,0 <= N,M <= 10000,所有字符串长度均在1至20之间,且均由小写字母构成.所有字典中的用户名互不相同,但是查询字符串可能有重复.