**旅行商(TSP)**

**Description**

Shrek is a postman working in the mountain, whose routine work is sending mail to n villages. Unfortunately, road between villages is out of repair for long time, such that some road is one-way road. There are even some villages that can’t be reached from any other village. In such a case, we only hope as many villages can receive mails as possible.

Shrek hopes to choose a village A as starting point (He will be air-dropped to this location), then pass by as many villages as possible. Finally, Shrek will arrived at village B. In the travelling process, each villages is only passed by once. You should help Shrek to design the travel route.

**Input**

There are 2 integers, n and m, in first line. Stand for number of village and number of road respectively.

In the following m line, m road is given by identity of villages on two terminals. From v1 to v2. The identity of village is in range [1, n].

**Output**

Output maximum number of villages Shrek can pass by.

**Example**

Input

4 3 1 4 2 4 4 3

Output

3

**Restrictions**

1 <= n <= 1,000,000

0 <= m <= 1,000,000

These is no loop road in the input.

Time: 2 sec

Memory: 256 MB

**Hints**

Topological sorting

**描述**

Shrek是一个大山里的邮递员，每天负责给所在地区的n个村庄派发信件。但杯具的是，由于道路狭窄，年久失修，村庄间的道路都只能单向通过，甚至有些村庄无法从任意一个村庄到达。这样我们只能希望尽可能多的村庄可以收到投递的信件。

Shrek希望知道如何选定一个村庄A作为起点（我们将他空投到该村庄），依次经过尽可能多的村庄，路途中的每个村庄都经过仅一次，最终到达终点村庄B，完成整个送信过程。这个任务交给你来完成。

**输入**

第一行包括两个整数n，m，分别表示村庄的个数以及可以通行的道路的数目。

以下共m行，每行用两个整数v1和v2表示一条道路，两个整数分别为道路连接的村庄号，道路的方向为从v1至v2，n个村庄编号为[1, n]。

**输出**

输出一个数字，表示符合条件的最长道路经过的村庄数。

**样例**

见英文题面

**限制**

1 ≤ n ≤ 1,000,000

0 ≤ m ≤ 1,000,000

**输入保证道路之间没有形成环**

时间：2 sec

空间：256 MB

**提示**

拓扑排序

**无线广播(Broadcast)**

**描述**

某广播公司要在一个地区架设无线广播发射装置。该地区共有n个小镇，每个小镇都要安装一台发射机并播放各自的节目。

不过，该公司只获得了FM104.2和FM98.6两个波段的授权，而使用同一波段的发射机会互相干扰。已知每台发射机的信号覆盖范围是以它为圆 心，20km为半径的圆形区域，因此，如果距离小于20km的两个小镇使用同样的波段，那么它们就会由于波段干扰而无法正常收听节目。现在给出这些距离小 于20km的小镇列表，试判断该公司能否使得整个地区的居民正常听到广播节目。

**输入**

第一行为两个整数n，m，分别为小镇的个数以及接下来小于20km的小镇对的数目。 接下来的m行，每行2个整数，表示两个小镇的距离小于20km（编号从1开始）。

**输出**

如果能够满足要求，输出1，否则输出-1。

**输入样例**

4 3

1 2

1 3

2 4

**输出样例**

1

**限制**

1 ≤ n ≤ 10000

1 ≤ m ≤ 30000

不需要考虑给定的20km小镇列表的空间特性，比如是否满足三角不等式，是否利用传递性可以推出更多的信息等等。

时间：2 sec

空间：256MB

**提示**

BFS

**重名剔除(Deduplicate)**

**Description**

Mr. Epicure is compiling an encyclopedia of food. He had collected a long list of candidates nominated by several belly-gods. As candidates in list are nominated by several people, duplication of name is inevitable. Mr. Epicure pay you a visit for help. He request you to remove all duplication, which is thought an easy task for you. So please hold this opportunity to be famous to all belly-gods.

**Input**

1 integer in fist line to denote the length of nomination list. In following n lines, each nomination is given in each line.

**Output**

All the duplicated nomination (only output once if duplication appears more multiple times), which is sorted in the order that duplication appears firstly.

**Example**

Input

10

brioche

camembert

cappelletti

savarin

cheddar

cappelletti

tortellni

croissant

brioche

mapotoufu

Output

cappelletti

brioche

**Restrictions**

1 < n < 6 \* 10^5

All nominations are only in lowercase. No other character is included. Length of each item is not greater than 40.

Time: 2 sec

Memory: 256 MB

**Hints**

Hash

**描述**

Epicure先生正在编撰一本美食百科全书。为此，他已从众多的同好者那里搜集到了一份冗长的美食提名清单。既然源自多人之手，其中自然不乏重复 的提名，故必须予以筛除。Epicure先生因此登门求助，并认定此事对你而言不过是“一碟小菜”，相信你不会错过在美食界扬名立万的这一良机

**输入**

第1行为1个整数n，表示提名清单的长度。以下n行各为一项提名

**输出**

所有出现重复的提名（多次重复的仅输出一次），且以其在原清单中首次出现重复（即第二次出现）的位置为序

**样例**

见英文题面

**限制**

1 < n < 6 \* 10^5

提名均由小写字母组成，不含其它字符，且每项长度不超过40

时间：2 sec

空间：256 MB

**提示**

散列

# THU2015 spring 3-1 TSP

### ****描述****

　　Traveling Salesman Problem

　　市场上有很多种商品，旅行商Shrek做短期买卖赚取差价。他从一个城市购买一件商品，到达下一个相邻的城市就卖掉。如果这一次买卖无利可图， 那么他就不会这么走。凭着自己和同伴多年的经验，他已经对地图上n个城市之间的差价了如指掌。两城市间可能有多种商品可赚取差价，此时Shrek只好倒卖 利润最大的一件商品。

　　现在请你帮Shrek规划一条路线，使他能赚最多的钱。

### ****输入****

　　第一行两个整数n、m。表示有n个城市，编号1~n

　　接下来有m行，每行三个整数a、b、price，表示从城市a到城市b可赚取差价price。

### ****输出****

　　若干Tab分隔的整数，连成一条赚钱最多的路线。若有多条，输出字典序最小的那条路线（编号小的城市靠前）

#### ****输入样例****

4 4

2 1 5

1 3 3

3 4 1

1 4 5

#### ****输出样例****

2 1 4

### ****限制****

　　2 <= n <= 100,000

　　1 <= m <= 1,000,000

　　差价范围是[1, 1000]

　　输入保证道路之间没有形成环。

　　时间：2 sec

　　空间：256MB

### ****提示****

#### ****一级提示****

　　拓扑排序

　　http://www.cnblogs.com/shanyou/archive/2006/11/16/562861.html

　　http://mindlee.net/2011/10/28/graph-search/

# THU2015 spring 3-2 Pokeface

### ****题目描述****

　　魔术师将一叠扑克顺次在桌上排成一行，全部正面朝上。之后的每一次挥一挥衣袖，都会翻转一连串的扑克，改变它们的正反朝向。从古代传下来的规 矩，魔术师皆有师傅，他们要随时准备回答师傅的问题。问题形如：目前从第i张扑克牌到第j张扑克牌中（包含第i和第j张扑克牌），有几张正面朝上？

　　现代的魔术师都有计算机，而他们的助手你被要求学习程序设计、算法与数据结构。现在他把这个问题交给了你，你又雇佣了一家公司为你做了前端识别 系统，通过置于魔术师表演桌面上面的摄像机捕捉到的影像，识别魔术师的挥一挥，并将挥一挥以标准形式传递给你的程序，你的程序在每一次挥一挥之后会维护相 应的数据结构来保存魔术师的桌上牌面情况，同时会开启一个监听进程对师傅提的问题进行语音识别，当然这个识别系统也是雇佣别的公司给你做的，对于每一个问 题，你要用尽量快的速度计算出答案，并通过蓝牙耳机传送给魔术师。

　　现在，你认为边边角角的手势识别系统和语音识别系统都已经交由公司解决了，那么核心的数据结构和问题计算必须由你自己亲自编写。

### ****输入****

　　第一行包含三个正整数n、m，其中n表示扑克牌的数量

　　接下来共m行，每行包含一个操作，操作分为2种：

H i j //魔术师挥一挥，改变了第i张牌到第j张牌的正反

Q i j //师傅问话：截至目前，第i张到第j张牌中有多少张正面朝上

　　H操作和Q操作的范围均包含边界（即第i张和第j张牌）。

　　对于每个操作，输入保证1 ≤ i ≤ j ≤ n。

### ****输出****

　　对于每次Q操作，输出一行，包含一个整数，表示问题的答案。

#### ****输入样例****

7 5

H 2 4

Q 1 6

H 3 6

Q 2 5

Q 1 7

#### ****输出样例****

3

2

4

### ****限制****

　　1 ≤ n ≤ 10^6

　　1 ≤ m ≤ 10^5

　　时间限制：1 sec

　　内存限制：256 MB

### ****提示****

#### ****一级提示****

　　线段树：http://baike.baidu.com/view/670683.htm

# THU2015 spring 3-3 Temperature

### ****描述****

某气象台每天都要从遍布于各地的观察站采集气温数据，并通过互联网为远程用户提供统计查询服务。其中最常见的一类查询是，根据用户指定矩形区域内所 有观察站的观测值计算出平均气温。随着更多观察站的不断建立，原始数据本身的规模急剧膨胀。另外，尽管可以假设每天采集的数据相对固定，但随着用户群体的 扩大，查询的频率也日益激增。鉴于传统蛮力算法的效率已无法满足实用要求，气象台只好请你帮忙，通过改进数据结构和算法，提高查询的效率。

借助气象台提供的一组函数接口，服务器端可访问已采集到的所有数据，并报告查询结果。

### ****接口说明****

int GetNumOfStation(void);

必须先调用该函数，返回观察站的总数n。

void GetStationInfo(int no, int \*x, int \*y, int \*temp);

获得第no个（0 <= no < n）观察站的信息：其地理坐标(\*x,\*y)及其所测温度值\*temp。各观测站的测量精度统一以0.01℃为基准单位，比如12.34℃表示为整数1234。

int GetQuery(int \*x1, int \*y1, int \*x2, int \*y2);

接收下一查询请求。返回值1对应于一次有效的查询。矩阵区域的四边分别与x或y轴平行，(\*x1,\*y1)和(\*x2,\*y2)分别为其西南角和东北角的坐标。恰好被矩形边界穿过的观察站，也视作落在其中。若返回0，则表示没有更多的查询，你的程序可以退出。

void Response(int temp);

针对当前的查询，在计算出对应的平均气温后，你可通过这一接口报告所得数值(截断取整，比如12.345℃输出为1234，-12.345℃输出为-1234)。

**特别注意**：每调用GetQuery()接收一次查询后，若未能通过Response()函数报告该次查询的结果 就再次调用GetQuery()接收下一查询，则将因为前次查询的结果无法报告而注定输出错误。也就是说，GetQuery()和Response()必 须交替调用。一个简单的查询示例程序如下：

int state;

do {

state = GetQuery(&x1, &y1, &x2, &y2);

// 计算 [x1, y1]-[x2, y2]范围内温度平均值, 存储到queryResult

Response(queryResult);

} while (state);

### ****测试说明****

为便于你调试和测试，随题还附带有[temperature.h](http://dsa.cs.tsinghua.edu.cn/oj/attachment/a131/a131e55c9a9c646547b1d4e08cc3a6407d8a77dc.h)和[temperature\_lib.c](http://dsa.cs.tsinghua.edu.cn/oj/attachment/786c/786c07be81c04dc9b831bc8219e554ee6a40d92f.c)文件。前者约定了上述接口，后者是这组接口的一种实现——OJ上的实现与之不同，但接口完全一致。调试时可将它们与你的代码一同编译，但在线测试时不必提交；即便提交，OJ也会自动忽略它们。

### ****输入****

脱机调试时，temperature\_lib.c所实现的三个输入接口，实际上是从当前目录下的temperature.in文件读入数据，因此通过按如下格式更改该文件，即可设定不同的输入数据：

第一行为两个整数：观察站总数n，所需查询的总次数m。

以下n行分别描述各观察站：位置坐标为整数(x, y)，该站所测得温度值为整数t。

再以下m行分别对应于各次查询操作，用整数(x1, y1)和(x2, y2)分别描述其西南角和东北角。

### ****输出****

脱机调试时，temperature\_lib.c所实现的Response()接口会在程序运行后，将所有的输出结果写入temperature.out文件。

文件共m行，各含1个整数，表示每次查询所得平均温度。

若查询区域不含任何观测站，则输出0。

#### ****输入样例****

4 2

0 0 1000

1 1 1300

2 2 1600

3 3 1100

0 0 1 1

0 0 10 10

#### ****输出样例****

1150

1250

### ****限制****

1 <= n <= 5\*10^4

1 <= m <= 5\*10^5

观测站坐标取值范围[-2^31, 2^31)

时间：10 sec

空间：256MB

### ****提示****

#### ****一级提示****

===\*\*一级提示\*\*===

　　先对所有观测站适当建立索引，再进行高效的查询。

　　可以使用Range Tree，你也可以选择其它数据结构。可参考讲义08.ABST.XC.More\_Search\_Trees、习题解析[8-20]。

# THU2015 spring 3-4 Hotname

### ****描述****

　　英文中的姓氏千奇百怪，为此Andy搜集了常见的姓氏，并且整合出了一套在线的姓氏查询引擎。

　　有一天，他突发奇想，希望通过引擎记录的日志来获得最热查询姓氏，即所有查询中被查询次数最多的姓氏。但是，为了不影响服务器的正常运行，Andy只能使用非常有限的内存资源来完成这项工作。

### ****输入****

　　第一行仅含一个整数，即查询日志中姓氏条目的总数n。

　　接下来的n行给出查询日志，每行各用一个字符串给出一个姓氏条目。

### ****输出****

　　仅一行，包含一个字符串（最热查询姓氏）和一个整数（对应的查询次数）。

#### ****输入样例****

10

bales

gazier

peterson

jones

watts

peterson

bales

wales

peterson

jones

#### ****输出样例****

peterson 3

### ****限制****

　　1 <= n <= 2\*10^6

　　姓氏总数 <= 10^4

　　所有被查询的姓氏均由小写字母组成，且长度不超过8个字符（不含换行符）。

　　最热的查询姓氏保证是唯一的。

　　时间：2 sec

　　空间：20 MB

　　使用的内存不得超过20MB，否则OJ会立即强制终止你的程序。

### ****提示****

#### ****一级提示****

　　提供两种可行的方法，你还可以使用其它方法

　　1. Hash

　　2. 字典树：http://zh.wikipedia.org/wiki/%E5%AD%97%E5%85%B8%E6%A0%91