

热点网站统计

题目描述：

企业路由器的统计页面，有一个功能需要动态统计公司访问最多的网页 $\text{URL}_{\text{top } N}$ 。请设计一个算法，可以高效动态统计 $\text{Top } N$ 的页面。

输入描述：

每一行都是一个 URL 或一个数字，如果是 URL ，代表一段时间内的网页访问；如果是一个数字 N ，代表本次需要输出的 $\text{Top } N$ 个 URL 。

输入约束：1、总访问网页数量小于 5000 个，单网页访问次数小于 65535 次；2、网页 URL 仅由字母，数字和点分隔符组成，且长度小于等于 127 字节；3、数字是正整数，小于等于 10 且小于当前总访问网页数；

输出描述：

每行输入要对应一行输出，输出按访问次数排序的前 N 个 URL ，用逗号分隔。

输出要求：1、每次输出要统计之前所有输入，不仅是本次输入；2、如果有访问次数相等的 URL ，按 URL 的字符串字典序升序排列，输出排序靠前的 URL ；

补充说明：

示例 1

输入：

news.qq.com

news.sina.com.cn

news.qq.com

news.qq.com

game.163.com

game.163.com

www.huawei.com

www.cctv.com

3

www.huawei.com

```
www.cctv.com
```

```
www.huawei.com
```

```
www.cctv.com
```

```
www.huawei.com
```

```
www.cctv.com
```

```
www.huawei.com
```

```
www.cctv.com
```

```
www.huawei.com
```

```
3
```

输出:

```
news.qq.com,game.163.com,news.sina.com.cn
```

```
www.huawei.com,www.cctv.com,news.qq.com
```

说明:

示例 2

输入:

```
news.qq.com
```

```
www.cctv.com
```

```
1
```

```
www.huawei.com
```

```
www.huawei.com
```

```
2
```

```
3
```

输出:

```
news.qq.com
```

```
www.huawei.com,news.qq.com
```

```
www.huawei.com,news.qq.com,www.cctv.com
```

说明:

```
import java.util.Scanner;
```

```
import java.util.*;
```

```
public class Main {

    public static void main(String[] args) {

        /*Scanner      cin      =      new      Scanner(System.in,

StandardCharsets.UTF_8.name());

        int num = cin.nextInt();

        int[] steps = new int[num];

        for (int i = 0; i < num; i++) {

            steps[i] = cin.nextInt();

        }

        cin.close();*/

        Scanner sc = new Scanner(System.in);

        TreeMap<String, Integer> map = new TreeMap<>();

        while (sc.hasNext()) {

            String s = sc.nextLine();

            String[] split = s.split(".");

            if(!s.contains(".")) {

                int n = Integer.parseInt(s);

                if(n > map.size()) {

                    break;

                }

                ArrayList<Map.Entry<String, Integer>> list = new

ArrayList<>(map.entrySet());
```

```
list.sort(new Comparator<Map.Entry<String, Integer>>() {
```

```
    @Override
```

```
        public int compare(Map.Entry<String, Integer> o1,
```

```
Map.Entry<String, Integer> o2) {
```

```
            return o2.getValue().compareTo(o1.getValue());
```

```
        }
```

```
    });
```

```
    String res = "";
```

```
    for (int i = 0; i < n; i++) {
```

```
        res += list.get(i).getKey();
```

```
        if(i != n - 1) {
```

```
            res += ",";
```

```
        }
```

```
    }
```

```
    System.out.println(res);
```

```
    } else {
```

```
        map.put(s, map.getDefault(s, 0) + 1);
```

```
    }
```

```
    }
```

```
}
```

```
}
```

