

c++-排序-企业路由器的统计页面

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 URL top N。请设计一个算法，可以高效动态统计 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 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
www.cctv.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:

```
#include <iostream>
```

```
#include <unordered_map>
```

```
#include <vector>
```

```
#include <algorithm>
```

```
using namespace std;
```

```
//输入: URL 数字 topN
```

```
//输出: 前 N 个
```

```
unordered_map<string,int> umap;
```

```
vector<pair<string,int>> vp;
```

```
bool cmp(const pair<string,int> &p1, const pair<string,int> &p2) {
```

```
    if(p1.second != p2.second) return p1.second > p2.second;
```

```
    else return p1.first < p2.first;
```

```
}
```

```
int main() {
```

```
    string str;
```

```
    while(getline(cin, str)) {
```

```
        if(str.find('.') != -1) {
```

```
            umap[str]++;
```

```
        }else {
```

```
            vp.clear();
```

```
            int top = stoi(str);
```

```
            for(auto it = umap.begin(); it != umap.end(); ++it) {
```

```
                vp.push_back(make_pair(it->first,it->second));
```

```
            }
```

```
            sort(vp.begin(),vp.end(),cmp);
```

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

```
                if(i < top - 1) cout << vp[i].first << ",";
```

```
                if(i == top - 1) cout << vp[i].first << endl;
```

```
            }
```

```
        }
```

```
        str = "";
```

```
    }
```

```
}  
// 64 位输出请用 printf("%lld")
```