

## 顺丰科技 2020校招 视觉算法工程师 在线考试

编程题 | 20.0分

1/2

### 计分板

时间限制: C/C++语言 1000MS; 其他语言 3000MS

内存限制: C/C++语言 131072KB; 其他语言 655360KB

题目描述:

假设你正在参加一场比赛, 这场比赛有A,B,C...M一共13题。现在, 你已经通过了n题, 此时你注意到, 大屏幕上有一个计分板, 里面记录了每一道题目目前有多少人通过。你想知道, 在剩下的还没有做出来题目当中, 哪一道题才是通过人数最多的题目? 如果有多个这样的题目, 取题号靠前的。

### 输入

第一行一个数n, 表示你现在通过的题目。接下来n个字母, 描述你通过的题号。接下来13个数, 表示每一道题有多少人通过。参加比赛的人数不大于150人

### 输出

输出一个字母, 代表在剩下的还没有做出来题目当中通过人数最多的题目。

### 样例输入

```
6
ADEGHM
58 42 98 84 84 40 75 97 98 7 8 40 54
```

### 样例输出

```
C
```

```
#include<iostream>
#include<algorithm>
#include"stdio.h"
#include<cstring>
#include<queue>
#include<vector>
#include<cmath>
#include<stack>
#include<deque>
#include<string>
#include <sstream>
#include <unordered_map>

using namespace std;

typedef pair<int, char> P;
bool cmp(const P p1, const P p2)
{
```

```

        if(p1.first > p2.first)
            return true;
        else if(p1.first == p2.first)
        {
            if(p1.second < p2.second)
                return true;
            else
                return false;
        }
        else
            return false;
    }

    const int MAX_N = 14;
    int pass[MAX_N];
    int times[MAX_N];
    P keyVal[MAX_N];

    int n;
    string str;
    /*
    6
    ADEGHM
    58 42 98 84 84 40 75 97 98 7 8 40 54
    */

    int main()
    {

        cin >> n;
        cin >> str;

        //priority_queue<P, vector<P>, cmp> que;
        priority_queue<P, vector<P>, less<P>> que;

        for(int i=0; i<n; i++)
            pass[int(str[i] - 'A')] = 1;

        for(int i=0; i<13; i++)
        {
            cin >> times[i];
            keyVal[i] = P(times[i], 'A' + i);
            //que.push(P(times[i], 'A' + i));
        }

        sort(keyVal, keyVal + 13, cmp);
        for(int i=0; i<13; i++)
        {
            P key = keyVal[i];
            if(!pass[int(key.second - 'A')])
            {
                cout << key.second << endl;
                break;
            }
        }

        //cout << keyVal[i].first << keyVal[i].second << endl;
    }

```

```
    return 0;
}
```

**时间限制：**C/C++语言 1000MS；其他语言 3000MS

**内存限制：**C/C++语言 131072KB；其他语言 655360KB

**题目描述：**

某学术会议上，一共有 $n$ 个人参加，现在已知每个人会的语言（一个人可能不会任何语言）。现在有一种学习机，每一个学习机可以在会议期间使一个人暂时掌握一种自己不会的语言，问要使得任意两人都要能直接或者间接的交流至少准备多少个学习机？

间接交流的意思是：可以通过其他参加会议的人翻译（可能或出现很多人一起帮忙翻译的情况）进行交流。如：第一个人和第二个人会第一种语言，第二个人和第三个人会第二种语言，那么第一个人可以和第三个人进行交流（通过第二个人的翻译）

#### 输入

第一行3个数 $n, m, k$ 代表人数，语言数，已知的信息数 接下来 $k$ 行，每行两个数 $u, v$ ，代表第 $u$ 个人会第 $v$ 种语言

#### 输出

输出需要准备的学习机的个数

#### 样例输入

```
3 3 2
2 3
3 1
```

#### 样例输出

```
2
```

#### 提示

**数据范围**

$1 \leq n \leq 100000, 1 \leq m \leq 100000, 0 \leq k \leq 100000, 1 \leq u \leq n, 1 \leq v \leq m$

```
#include<iostream>
#include<algorithm>
#include"stdio.h"

using namespace std;

const int MAX_N = 210000;
int f[MAX_N];
int vis[MAX_N];

int fa(int a)
```

```

{
    if(a == f[a])
        return a;
    else
        return f[a]=fa(f[a]);
}

int n, m, k;
/*
3 3 2
2 3
3 1
*/

int main()
{
    cin >> n >> m >> k;

    for(int i=1; i<=m+n; i++)
    {
        f[i]=i;
        vis[i]=-1;
    }

    for(int i=1; i<=k; i++)
    {
        int u, v;
        cin >> u >> v;
        v += n;
        f[fa(u)] = fa(v);
    }

    int ans=0;
    for(int i=0; i<=n; i++)
        if(vis[fa(i)] == -1)
        {
            vis[fa(i)] = 1;
            ans++;
        }
    if(k==0)
        cout << ans << endl;

    else
        cout << ans - 1 << endl;

    return 0;
}

```

