

最大剩余法

评判的指标

$$m = q * \frac{p}{N}$$

m: 表示某单位的席位数

q: 表示总席位数

p: 某单位的人数

N: 代表总人数

因为 m 可能是小数表示的，但是保证每一个单位都可以取到 m 的整数部分的席位数。假设还剩下rest个席位，对每一个单位的小数部分进行排序，取前rest个单位，给他们一个席位。

实现的代码

```
#include <bits/stdc++.h>
using namespace std;
const int total_number = 1000;
const int chair = 10;
struct Group{
    string group_name;
    int number;
    int integer;
    double rat;
    int final_chair;
}group[5];
bool cmp(Group a, Group b){
    return a.rat>b.rat;
}
bool cmp1(Group a, Group b){
    return a.group_name<b.group_name;
}

int main()
{
    int people_number[3] = {235, 333, 432};
    int temp = 0;
    for(int i=0; i<3; i++){
        group[i].group_name = char('a'+i);
        group[i].number = people_number[i];
```

```

        group[i].integer = int(1.0*people_number[i]/total_number*10);
        //cout<<group[i].integer<<endl;
        group[i].rat = double(people_number[i])/total_number*10.0-
group[i].integer;
        group[i].final_chair = group[i].integer;
        temp += group[i].integer;
    }
    sort(group, group+3, cmp);
    int rest = chair - temp;
    for(int i=0; i<rest; i++){
        group[i].final_chair++;
    }
    sort(group, group+3, cmp1);
    for(int i=0; i<3; i++){
        cout<<group[i].group_name<<": "<<group[i].final_chair<<endl;
    }
    return 0;
}

/*
output:
a:3
b:3
c:4
*/

```

最大 Q 值法

评判的指标

经过 ppt 上面一系列的运算、化简，得到下面的评判的标准式：

$$Q_i = \frac{p_i^2}{n_i * (n_i + 1)}$$

其中 p_i 表示的是单位 i 的人数， n_i 表示的是这个单位目前的席位数目。

初始下面，每一个单位都会被分配一个席位，然后计算每一个单位的 Q 值，然后选择 Q 值最大的单位，给他们单位分配一个席位。

如此下去，不断的对 Q 进行排序，直到分配完所有的席位。

实现的代码

```
#include <bits/stdc++.h>
using namespace std;
const int total_number = 1e3;
const int chair = 10;
const int group_number = 3;
struct Group{
    string name;
    int number;
    double q_value;
    int final_chair;
    Group(int number=0, double q_value = 0, int final_chair = 1)
        :number(number),q_value(q_value),final_chair(final_chair){}
}group[5];

bool cmp(Group a, Group b){
    return a.q_value>b.q_value;
}

bool cmp1(Group a, Group b){
    return a.name<b.name;
}

int main()
{
    int number[3] = {235, 333, 432};
    for(int i=0; i<group_number; i++){
        group[i].name = char('a'+i);
        group[i].number = number[i];
        group[i].q_value = number[i]*number[i]/
            (group[i].final_chair*(group[i].final_chair+1));
    }
    int rest = chair - group_number;
    for(int i=0; i<rest; i++){
        sort(group, group+group_number, cmp);
        group[0].final_chair++;
        group[0].q_value = group[0].number*group[0].number/
            (group[0].final_chair*(group[0].final_chair+1));
    }
    sort(group, group+group_number, cmp1);
    for(int i=0; i<group_number; i++){
        cout<<group[i].name<<" : "<<group[i].final_chair<<endl;
    }
}
```

```
    return 0;  
}  
/*  
output:  
a: 2  
b: 3  
c: 5  
*/
```