**★★★☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：12390: Distributing Ballot Boxes**

**解題者：陳欣愉**

**解題日期：2025年5月8日**

**題意：**

西班牙全國大選需要在各城市設置投票箱（ballot boxes)。

已知全國有N個城市(1 ≤ N ≤ 500, 000) ，B個投票箱(N ≤ B ≤ 2, 000, 000)，每個城市至少有一個箱子，每個城市有固定人口數ai , (1 ≤ ai ≤ 5, 000, 000) 。在選民不能跨越城市投票的狀況下，目標設計出「箱子分配方式」，使得「最擁擠的箱子人數」最小。

。

**題意範例：**

Input：

 2 7 // 城市總數 , box數

 200000 // 城市A人口數

 500000 // 城市B人口數

 4 6 // 城市總數 , box數

 120 // 城市a人口數

 2680 // 城市b人口數

 3400 // 城市c人口數

 200 // 城市d人口數

 -1 -1 // end of input

Output：

 100000 // 容量最大的票箱分配人口數

 1700

|  |  |  |  |
| --- | --- | --- | --- |
|  | 城市 A(200000人) | 城市 B(500000人) |  |
| 分配箱數 | 2 | 5 | 總箱數 7 |
| 每箱人數 | 100000 | 100000 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 城市 a(120人) | 城市 b(2680人) | 城市 c(3400人) | 城市 d(200 人) |  |
| 分配箱數 | 1 | 2 | 2 | 1 | 總箱數 6 |
| 每箱人數 | 120 | 1340 | 1700 | 200 |  |

**解法：**

1. 先猜測一個箱子容量上限s，假設「不可以超過s=3000人一箱」。將

該組測資中城市人口數最大值設為s軸最右側邊界，作為測試起始。

1. 檢查每個城市需要幾個箱子才能讓每箱<=3000
2. 計算出總共需要幾個箱子
3. 如果使用的箱子數<=題目提供的箱子總數，則猜測值s是可行的方案。
4. 透過Binary Search夾擠找出最佳解。

**解法範例：**

城市 A : 120 人

城市 B : 2680 人

城市 C : 3400 人

城市 D : 200 人

共有4座城市，提供6個投票箱。

* 城市人口最大值為3400人，設置為初始測試值。
* 假設一箱最多可容納3400人

城市A: 需要1箱

城市B: 需要1箱

城市C: 需要1箱

城市D: 需要1箱

共需4箱$\leq $6，可行方案。

* 假設一箱最多可容納119人

城市A: 需要2箱

城市B: 需要23箱

城市C: 需要29箱

城市D: 需要2箱

共需56箱不$\leq $6，不可行方案。

* 假設一箱最多可容納1759人🡪(3400+119)/2 自動捨去小數點後位數

城市A: 需要1箱

城市B: 需要2箱

城市C: 需要2箱

城市D: 需要1箱

共需6箱不$\leq $6，可行方案。

**可行區間**

1759

可行

mid

119

不可行

Left

3400

可行

Right

**可行區間**

1759

可行

Right

119

不可行

Left

依序二元搜索下去持續縮小範圍，直到最後在1700處找到最佳解。

**討論：**

* 取mid方式不同，會造成移動left,right 標記不同。

本題區間 : Left < s $\leq $ Right

因此使用mid = (left+right) /2 取值偏左的形式。

假使 mid =(left+right+1)/2 則會造成取值偏右的狀況。

* 計算箱數要取上限，因為人口可能不會剛好被箱數整除。

城市人口 val = 10人

假設每箱最多塞 s = 3人

10/3=3.3333 🡪 3箱。但實際應該分配4個箱子，因為不能有人沒被分到箱子。

[3][3][3][1]

val / s 🡪 c++會自動刪去小數點後，取下限

( val+s-1)/s 🡪 即取上限 (加s再除以s會進1，但邊界要排除，所以加少一點s-1)

**程式：**

# include <bits/stdc++.h>

using namespace std;

bool check(vector<int> &v, int b, int s){

 int count=0;

 for(int val: v){

 int k= (val+s-1)/s;

 count +=k;

 }

 return count<=b;

}

int main(){

 int n, b;

 while(cin>>n>>b && n!=-1 && b!=-1){

 vector<int> v(n);

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

 cin >> v[i];

 }

 int left=0, right=5000000;

 while(left<right){

 int mid = (left+right)/2;

 if(check(v, b, mid)){

 right = mid;

 }

 else{

 left = mid+1;

 }

 }

 cout << left << endl;

 }

 return 0;

}