1320 均分纸牌

题目描述

有n堆纸牌，编号分别为 1，2，…,n。每堆上有若干张，但纸牌总数必为n的倍数。可以在任一堆上取若干张纸牌，然后移动。

移牌规则为：在编号为1的堆上取的纸牌，只能移到编号为 2 的堆上；在编号为 n 的堆上取的纸牌，只能移到编号为n-1的堆上；其他堆上取的纸牌，可以移到相邻左边或右边的堆上。

现在要求找出一种移动方法，用最少的移动次数使每堆上纸牌数都一样多。

例如 n=4，4堆纸牌数分别为： ① 9 ② 8 ③ 17 ④ 6移动3次可达到目的：

从③取4张牌放到④（9 8 13 10）->

从③取3张牌放到②（9 11 10 10）->

从②取1张牌放到①（10 10 10 10）。

输入

n（n 堆纸牌，1≤n≤100）

a1a2…an（n 堆纸牌，每堆纸牌初始数，1≤ai≤10000）。

输出

所有堆均达到相等时的最少移动次数。

输入样例

4
9 8 17 6

输出样例

3

解析

逻辑航线 微信号：luojihangxian

最靠边的两个牌堆移动受限，中间的牌可以自由移动。并且以每次移动的牌数来达到平均数，也就是每堆牌把平均数当成目标去靠近。

贪心：一开始算出平均数，然后从左往右以这个平均数为目标匀，所以可能只要匀一圈就可以成功；

计算过程：一开始就算出平均数，每堆牌和平均数比较，用平均数减每堆的牌数，剩下的牌数就是离平均数的“距离”，然后从左往右匀=》不管这堆牌的正负，让此堆牌加到下一堆牌，然后再让此堆牌为零。

证明：如果每堆牌的目标不是平均值，则需要移动更多的次数。

样例：

（9+8+17+6）/4=10；

对应贪心的牌数为：-1；-2；7；-4；

第一次匀：0；-3；7；-4；

第二次：0；0；4；-4；

第三次：0；0；0；0；

编码

```
using namespace std;

int pile[105];   // 记录初始状态

int main() {
    int N;
    cin >> N;

    //记录输入的纸牌信息
   int tot = 0;
    for (int i = 0; i < N; i++) {
        cin >> pile[i];
        tot += pile[i];
    }
    //计算出每堆的平均数
   int avg = tot / N;

    int count = 0;
    // 开始模拟移动的过程
   for (int i = 0; i < N; ++i) {
         // 如果当前堆纸牌数大于avg，那就将当前多余的纸牌数移到后一堆里，
       // 如果小于，就将后一堆纸牌多余的移到当前堆里，同时做计数
```

```
        // 事实上这里不排斥pile[i + 1]变为负数的情况，但正确性仍然是可以保证的
        if (pile[i] != avg) {
            //无论正负，就是移动
            pile[i + 1] += pile[i] - avg;
            count++;
        }
    }

    cout << count << endl;
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

逻辑航线 微信号：luojihangxian