



# Luogu P1031 均分纸牌



作者

ProtectEMmm

发布时间

2024-08-30 15:45

分类

题解 P1031

## Luogu P1031 均分纸牌

### 题意

给定  $n$  堆纸牌，第  $i$  堆纸牌有  $a_i$  张，保证  $\sum_{i=1}^n a_i$  是  $n$  的倍数。

可以进行一种操作：选取相邻的两堆，从某一堆中拿任意张牌给另一堆，算一次操作。

求最小的操作次数，使得  $n$  堆牌中牌的数量都一样。

### 题解

大部分题解只讲了如何贪心，并没有讲贪心的正确性证明。导致我在做这题时，包括很多选手做这题时产生了很多疑惑。今日特此补上。

首先先要提出一点，这题不允许牌堆是负数。所以题解里基于贪心的贷款思路从题意上来说，是错误的。但是我们可以证明最后这么做不影响正确性。

首先很容易看出来，最后要求所有牌堆的牌数都等于平均数。

**引理：** 任意两堆相邻的牌堆，最优策略只能从一堆往另一堆一次性给出若干张牌。不存在  $A$  给了  $B$ ，之后  $B$  又给了  $A$ 。

我们可以用分治的思想证明这个引理。

首先任取两堆相邻的牌，不妨设左边所有牌堆的总和为  $A$ ，右边所有牌堆的总和为  $B$ 。

因为我们要求每堆牌最后都相等，所以可以让两边都减去每堆牌至少需要的牌数。

现在有三种情况。

1.  $A = B = 0$ ：说明两堆之间不需要进行牌的交换。
2.  $A > 0 > B$ ：说明  $A$  有余， $B$  不足。
3.  $A < 0 < B$ ：说明  $A$  不足， $B$  有余。

不妨设  $A$  有余， $B$  不足。反之同理。

那么肯定是  $A$  一次性把这么多牌给  $B$ 。

这里 小粉兔 提出了一个疑问：凭什么一定是 A 一次性把这么多牌给 B。（玄学一点：天之道，损有余而补不足）

这里我并不能证明一定是 A 一次性给 B。 小粉兔 紧跟着提问：为什么不能 B 给 A，然后再给 B。（先富带动后富）

这个提问给了我一点提示，（有一个比较玄学的回答是奥卡姆剃刀原理，这么多步骤肯定不合理），这里证明方法是，因为我们存在操作步骤次数为 1 的方案，所以其余方案都不优。从最优方案的存在性上下手。

现在我们证明了 A 和 B 之间一定是有余给不足，至于左侧牌堆内部，和右侧牌堆内部怎么处理，继续递归下去（或者用数学归纳法证明）。

如果我们考虑，有余往不足的牌堆连边。我们会发现问题变成了一张  $n$  点，至多  $n - 1$  条边的有向图。（为什么是至多，因为有可能两边互相都不需要给牌）

如果这张有向图是一张 DAG，即这张有向图无环。我们就可以按照拓扑序来进行给牌操作。这样就不会出现中间是负数的情况了。

现在如何证明这张有向图无环呢？一个非常不严谨的证明是， $n$  点，至多  $n - 1$  条边的有向图是一棵树，不可能有环。

问题是，如果有一些点本身就等于平均值，不需要连边。那就释放出来了本该连给他，现在多余的边，这些多余的边到其他地方会不会形成环呢？

如果我们释放了一些点，那么这些点就可以从  $n$  个点里忽略掉。所以最后还是  $n'$  点， $n' - 1$  边。

另一个解释是，这张图只能是一条链，因为不存在双向边所以绝不可能有环。

事实上，这张图最后形态是有若干条链。答案就是点数减去链数，因为不同链之间不需要进行交换，所以减掉。因为链是树的一种，可以看成是一个森林。那点数减链数其实就是边数。边数本质就是操作的次数。

我们发现我们最后只关心边的数量，不关心边的方向了。所以这题用贪心的思路做后，可以存在负数，可以贷款，这不影响这题的正确性。

这题如果没考虑这么多，只考虑贪心做的话。

第一堆牌和第二堆牌就三种可能：

1. 第一堆牌刚好：不需要处理。
2. 第一堆牌有余：只能给第二堆牌。
3. 第一堆牌不足：只能第二堆牌给。

于是我们就处理完了第一堆牌。也就是第一块多米诺骨牌已经推倒了。

对于第二堆牌来说，第一堆牌已经处理完了，不需要再考虑了，于是第二堆牌不能向左再操作了，只能向右。第二堆牌变成了第一堆牌。

于是就像多米诺骨牌一样，全部倒下去（也就是数学归纳法）。

补充：这题有一个环形版本，[负载均衡问题](#)，是[网络流24题](#)中的一道费用流题目。但是也可以像这题一样做。我们只需要证明这题一定能找到一个断点使他变成[均分纸牌](#)即可。

## 代码

```
#include<bits/stdc++.h>
using namespace std;
/*=====*/
#define endl "\n"
/*=====*/
typedef long long lnt;
/*=====*/
const int N = 1e2 + 10;
/*=====*/
int n, arr[N];
/*=====*/
void Solve(void)
{
    cin >> n;
    int average = 0;
    for (int i = 1; i <= n; ++i)
    {
        cin >> arr[i];
        average += arr[i];
    }
    average /= n;
    /*=====*/
    int ans = 0;
    for (int i = 1; i + 1 <= n; ++i)
    {
        if (arr[i] != average)
        {
            ans++, arr[i + 1] += arr[i] - average;
        }
    }
    cout << ans << endl;
    /*=====*/
    //第二份代码的思路就是，统计分治证明里提到的 A == B的断边。
    int ans = n, sum = 0;
    for (int i = 1; i <= n; ++i)
    {
        sum += arr[i] - average;
        if (sum == 0)ans--;
    }
    cout << ans << endl;
}
/*=====*/
int main()
{
    #ifndef ONLINE_JUDGE
        freopen("IN.txt", "r+", stdin);
    #endif
    ios::sync_with_stdio(false);
    cin.tie(NULL), cout.tie(NULL);
}
```

```
int T = 1; //cin >> T;
while (T--)Solve();
return 0;
}
```

致谢

感谢 Cherishdevil 在菜群提出了这个问题，使得我能够再次注意到这道题。

感谢 SkyWave 提出我可以针对这道题重新写一篇题解试试能不能让管理挂上。

感谢 小粉兔 在群内和我讨论，把我思路中的错误给指出，并不断给出新提示。

作者：ProtectEMmm 创建时间：2024-08-30 15:45:05

  
1

  
7

  
不推荐


评论区

发表评论

发表

1 条评论

默认排序

 **CyansQwQ** 回复于 4 个月前

这是我敬爱的群主吗

加载更多