1. 采用贪心算法，集合A表示付款数集合，集合B表示收款数集合。
2. 对冲抵消全部相等的结果，相加抵消
3. 从小到大排序，对冲抵消
4. 先抵消大的，若>,则直接抵消重新排序
5. 若<,则先作差，从集合找找等于差值或是大于差值的，做抵消，剩余的数再回到集合中
6. 重复2-4

贪心原理：相等的数抵消后，集合A（付款）和集合B（收款）中不再有相等的数，相等的数抵消对于一个数来说是最优的结果。

集合B中某个数为集合A中几个数的和，那么集合A中这几个数可以满足最优。这种几个数加起来等于其它一个数，这和算24点类似。关于这一点是否可以换一个思路，相当于是优先拆B中的数据？。

Step1:如果剩余的数中,那么是必须要拆开的，此时将拆为和两个数，抵消后余下tmp，在集合A中被删除，tmp回到A集合中，从集合B中删除，A,B集合均得到更新

情况1：更新后，B集合中有数和tmp相等，那么tmp再次抵消，对于一个必须拆开的数来说，这是最优的。

情况2：更新后，B集合中有没有数和tmp相等，那么tmp继续留在A集合中，等待后续操作中是否会有与tmp相等的数与他抵消。此处再说明情况，如果step1之前，集合A中就有和tmp相等的数，那么他们的地位都是相同的，都是需要被抵消或是被拆分的，有抵消时为了维持一个数尽量被拆的次数少原则，此时，在有数可以被抵消时，tmp应该有更高的优先级。如果和tmp相等的数也是被拆开来的，他们应该按照各自优先级来操作。

如果到tmp变为序列A中最大值时，且在和tmp相等的数中tmp的优先级最高，序列B中没有可以和tmp抵消的，那么tmp需要继续被拆分。