1000 pt.

这是一个subset sum问题的变种，需要开发一个伪多项式时间的算法。

首先需要推导出任意一种分组S，T，其最大差别为：

也即max(tB-s,s-tA)，可见只跟s相关。

记f(p,s)为已经选定了从[0,p-1]的元素，组成了现在的s，在这种情况下最小的diff。

如果是这样，那么对p有两种选择：

1. 选择p，则最小选择为f(p+1,s+A[p]+B[p])
2. 不选p，则最小选择为f(p+1,s)
3. 如果p==n，则最小选择为固定的：max(tB-s,s-tA)

这样就可以用memo来计算，但是由于n\*max(s)很大，所以用memo会爆内存。故考虑使用DP。

按照DP的思想，应按照递推关系先计算最终结果，再向前递推：

dp[n][s]=max(tB-s,s-tA)

dp[p][s]=

1. Dp[p+1][s]
2. Dp[p+1][s+A[p]+B[p]]

如此计算。