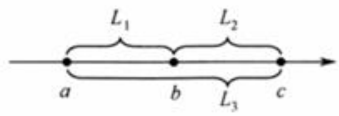
分析。由 $D=|a-b|+|b-c|+|c-a|\geqslant 0$ 得：

① 当 $a=b=c$ 时，距离最小。

② 其余情况。不失一般性，假设 $a\leqslant b\leqslant c$，观察下面的数轴：



$L_1=|a-b|$，$L_2=|b-c|$，$L_3=|c-a|$，$D=|a-b|+|b-c|+|c-a|=L_1+L_2+L_3=2L_3$

由 D 的表达式可知，事实上决定 D 大小的关键是 a 和 c 之间的距离，于是问题就可以简化为每次固定 c 找一个 a 使得 $L_3=|c-a|$ 最小。

1）算法的基本设计思想

① 使用 $D_{\min}$ 记录所有已处理过的三元组的最小距离，初值为一个足够大的整数。

② 集合 S_1、S_2 和 S_3 分别保存在数组 A、B、C 中。数组的下标变量 $i=j=k=0$，当 $i<|S_1|$、$j<|S_2|$ 且 $k<|S_3|$ 时（$|S|$ 表示集合 S 中的元素个数），循环执行 a)～c)。

a) 计算（A[i], B[j], C[k]）的距离 D；（计算 D）

b) 若 $D<D_{\min}$，则 $D_{\min}=D$；（更新 D）

c) 将 A[i], B[j], C[k]中的最小值的下标+1；（对照分析：最小值为 a，最大值为 c，这里 c 不变而更新 a，试图寻找更小距离 D）

③ 输出 $D_{\min}$，结束。