

11.15.1 推销员 (salesman)

【题目描述】

螺丝街是一条死胡同，出口与入口是同一个，街道的一侧是围墙，另一侧是住户。螺丝街一共有N家住户，第i家住户到入口的距离为Si米。由于同一栋房子里可以有多家住户，因此可能有多家住户与入口的距离相等。小光会从入口进入，依次向螺丝街的X家住户推销产品，然后按原路走出去。

小光每走1米就会积累1点疲劳值，向第i家住户推销产品会积累Ai点疲劳值。小光想知道，对于不同的X家住户，在不走多余的路的前提下，他最多会积累多少点疲劳值。

【输入格式】

第1行有一个正整数N，表示螺丝街住户的数量。

第2行有N个正整数，其中第i个正整数Si表示第i家住户到入口的距离。数据保证 $S_1 \leq S_2 \leq \dots \leq S_N < 10^8$ 。

第3行有N个正整数，其中第i个正整数Ai表示向第i家住户推销产品会积累的疲劳值。数据保证 $A_i < 10^3$ 。

【输出格式】

输出M行，每行一个正整数，第i行的正整数表示当X=i时，小光最多积累的疲劳值。

【输入样例1】

5
1 2 3 4 5
1 2 3 4 5

【输出样例1】

15
19
22
24
25

【样例说明1】

X=1：向住户5推销，往返走路的疲劳值为5+5，推销的疲劳值为5，总疲劳值为5+5+5=15。
X=2：向住户4、5推销，往返走路的疲劳值为5+5，推销的疲劳值为4+5，总疲劳值为5+5+4+5=19。
X=3：向住户3、4、5推销，往返走路的疲劳值为5+5，推销的疲劳值为3+4+5，总疲劳值为5+5+3+4+5=22。
X=4：向住户2、3、4、5推销，往返走路的疲劳值为5+5，推销的疲劳值为2+3+4+5，总疲劳值为5+5+2+3+4+5=24。
X=5：向住户1、2、3、4、5推销，往返走路的疲劳值为5+5，推销的疲劳值为1+2+3+4+5，总疲劳值为5+5+5+1+2+3+4+5=25。

【输入样例2】

5
1 2 2 4 5
5 4 3 4 1

【输出样例2】

12
17
21
24
27

【样例说明2】

X=1：向住户4推销，往返走路的疲劳值为4+4，推销的疲劳值为4，总疲劳值为4+4+4=12。
X=2：向住户1、4推销，往返走路的疲劳值为4+4，推销的疲劳值为5+4，总疲劳值为4+4+5+4=17。
X=3：向住户1、2、4推销，往返走路的疲劳值为4+4，推销的疲劳值为5+4+4，总疲劳值为4+4+5+4+4=21。
X=4：向住户1、2、3、4推销，往返走路的疲劳值为4+4，推销的疲劳值为5+4+3+4，总疲劳值为4+4+5+4+3+4=24。或者向住户1、2、4、5推销，往返走路的疲劳值为5+5，推销的疲劳值为5+4+4+1，总疲劳值为5+5+5+4+4+1=24。
X=5：向住户1、2、3、4、5推销，往返走路的疲劳值为5+5，推销的疲劳值为5+4+3+4+1，总疲劳值为5+5+5+4+3+4+1=27。

【数据范围】

对于100%的数据， $1 \leq N \leq 100\,000$ 。

【算法分析】

使用贪心算法，贪心算法是指在对问题求解时，总是做出在当前看来最好的选择。

首先选择疲劳值最大的一个点，即 $s[i] \times 2 + A[i]$ 值最大的点，将这个点记为now，用优先队列保存now左边的点的疲劳值。每次选择点时，枚举now右边的点，找出now右边疲劳值最大的点记为Rmax（注意这里的距离应该为now到的距离），与优先队列的最大值Lmax比较，取最大值的点为新的now。如果左边的点的疲劳值大，则将该点弹出优先队列；如果右边的点的疲劳值大，则将原先now点到新now点之间的点压入优先队列。时间复杂度在最好的情况下是 $O(n\log n)$ ，在最坏情况下是 $O(n^2)$ 。具体情况如图11.10所示。

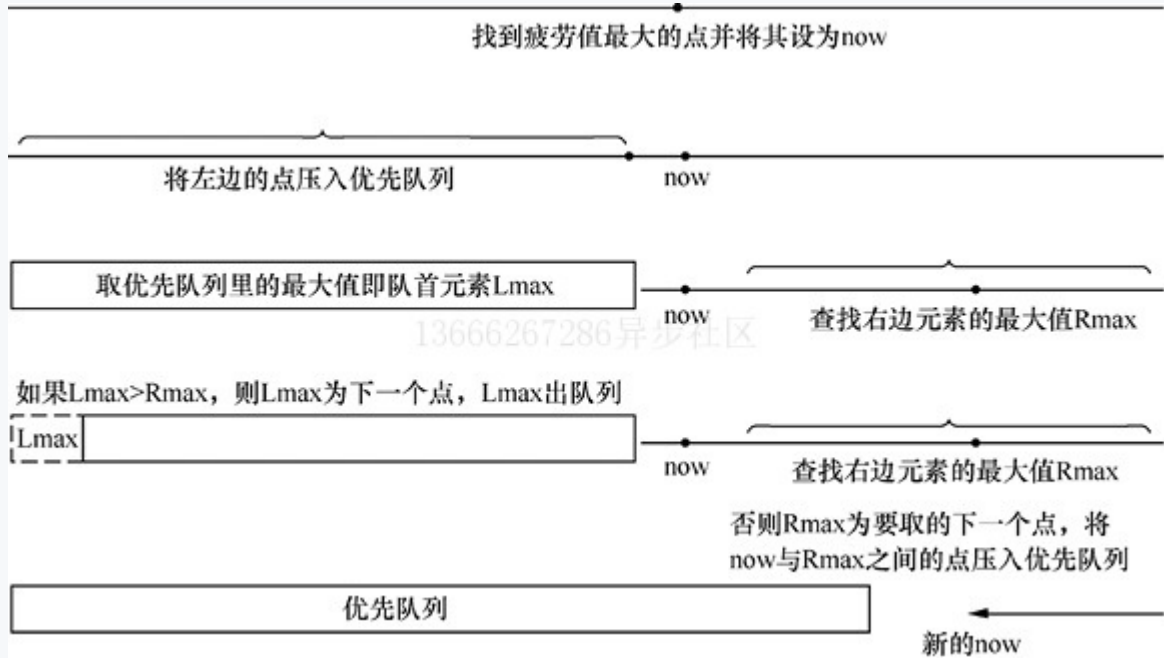


图11.10