**128MB,1S,.xxx**

最优二叉检索树

**问题描述**

你要建立一个二叉检索树，其上共有n个元素（编号1到n），已知每个元素会被访问的次数。每个元素每被访问一次的代价为其深度，我们认为树的根结点深度为1。任意一个子节点(child)的深度是其亲结点（parent）的深度+1。

我们认为编号小的元素，严格小于编号大的元素。

我们要求任意一个结点x都满足：x左子树的所有元素都严格小于x，x右子树的所有结点都严格大于x。

我们希望你设计一种安排方式，使得总的代价和最小。

若有多种方案，请输出先序遍历时字典序最小的方案。

对于两个方案，从第一个位置开始依次比较两个方案对应位置的数，假设第一个不同的位置为i，则第i个数（而不是字符/数字）小的方案字典序更小。

题面中的“访问”的代价已经将对其祖先的遍历的代价抽象为其深度，无需重复计算。

**输入描述**

第一行一个整数n表示元素个数

第二行n个整数，按照编号1到n的顺序给出这些元素的被访问次数。

**输出描述**

第一行一个整数，表示在最优情况下，总的代价。

第二行n个整数，表示最优的情况下，字典序最小的方案的先序遍历。

**样例输入**

**样例输出**

**数据范围及提示**