

Задача А. Дерево отрезков

Имя входного файла:	build-tree.in
Имя выходного файла:	build-tree.out
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Дан массив a длины n (нумерация массива a начинается с нуля). Требуется построить дерево отрезков (с операцией минимума) на этом массиве.

Дерево отрезков на массиве из n элементов — это бинарное подвешенное дерево, отвечающее следующим свойствам:

- Каждой вершине соответствует некий полуинтервал $[L_i; R_i)$.
- Вершина является листом тогда и только тогда, когда $R_i - L_i = 1$
- Если вершина не является листом, то ее левому и правому сыновьям соответствуют полуинтервалы $[L_i; \lfloor \frac{L_i+R_i}{2} \rfloor)$ и $[\lfloor \frac{L_i+R_i}{2} \rfloor; R_i)$ соответственно.
- Корню дерева соответствует полуинтервал $[0; N)$.

Значением в вершине дерева, которой соответствует полуинтервал $[L_i; R_i)$, является минимальное среди чисел $a_{L_i}, \dots, a_{R_i-1}$.

Вершинам дерева присваиваются номера по следующему принципу:

- Корень имеет номер 1.
- Левый и правый сын вершины с номером i имеют номера $2i$ и $2i + 1$ соответственно.

Формат входных данных

В первой строке вводится натуральное число $n \leq 30000$ — длина массива. Во второй строке вводится n чисел, не превышающих по модулю 10^5 .

Формат выходных данных

В первой строке выведите число m - количество вершин в дереве.

В следующих m строках выведите описание вершин дерева. В $i + 1$ -ой строке выведите шесть чисел, описывающих i -ую вершину дерева, через пробелы: номер i -ой вершины в дереве, границы полуинтервала L_i и R_i , значение в i вершине, а также номера вершин — левого и правого сына i -ой вершины (если у вершины нет сыновей, выведите два числа -1).

Левый и правый сыновья (если они есть) вершины с номером i , должны иметь номера $2i$ и $2i + 1$ соответственно. Корень должен иметь номер 1.

Требуется выводить вершины в порядке возрастания номеров.

Примеры

build-tree.in	build-tree.out
6 -1 3 4 3 -1 2	11 1 0 6 -1 2 3 2 0 3 -1 4 5 3 3 6 -1 6 7 4 0 1 -1 -1 -1 5 1 3 3 10 11 6 3 4 3 -1 -1 7 4 6 -1 14 15 10 1 2 3 -1 -1 11 2 3 4 -1 -1 14 4 5 -1 -1 -1 15 5 6 2 -1 -1
3 6 1 5	5 1 0 3 1 2 3 2 0 1 6 -1 -1 3 1 3 1 6 7 6 1 2 1 -1 -1 7 2 3 5 -1 -1