

Задача А. Хип ли?

Имя входного файла: `isheap.in`
Имя выходного файла: `isheap.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Структуру данных *Heap* можно реализовать на основе массива.

Для этого должно выполняться *основное свойство Heap'a*, которое заключается в следующем. Для каждого $1 \leq i \leq n$ выполняются следующие условия:

- Если $2i \leq n$, то $a[i] \leq a[2i]$
- Если $2i + 1 \leq n$, то $a[i] \leq a[2i + 1]$

Дан массив целых чисел. Определите является ли он *Heap'ом*.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка содержит n целых чисел по модулю не превосходящих $2 \cdot 10^9$.

Формат выходного файла

Выведите «YES», если массив является *Heap'ом* и «NO» в противном случае.

Примеры

<code>isheap.in</code>	<code>isheap.out</code>
5 1 0 1 2 0	NO
5 1 3 2 5 4	YES

Задача В. Хипуй!

Имя входного файла: `heap.in`
Имя выходного файла: `heap.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

В этой задаче вам необходимо организовать структуру данных *Heap* для хранения целых чисел, над которой определены следующие операции:

- `Insert(X)` — добавить в *Heap* число X ;
- `Extract` — достать из *Heap* наибольшее число (удалив его при этом).

Формат входного файла

Во входном файле записано количество команд N ($1 \leq N \leq 100\,000$), потом последовательность из N команд, каждая в своей строке.

Каждая команда имеет такой формат: „0 <число>“ или „1“, что означает соответственно операции `Insert(<число>)` и `Extract`. Добавляемые числа находятся в интервале от 1 до 10^7 включительно.

Гарантируется, что при выполнении команды `Extract` в структуре находится по крайней мере один элемент.

Формат выходного файла

В выходной файл для каждой команды извлечения необходимо вывести число, полученное при выполнении команды `Extract`.

Примеры

<code>heap.in</code>	<code>heap.out</code>
7	100
0 100	50
0 10	
1	
0 5	
0 30	
0 50	
1	

Задача С. Быстрая сортировка

Имя входного файла: `sort.in`
Имя выходного файла: `sort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан массив целых чисел. Ваша задача — отсортировать его в порядке неубывания с помощью алгоритма быстрой сортировки (`qsort`).

Формат входного файла

В первой строке входного файла содержится число N ($1 \leq N \leq 100\,000$) — количество элементов в массиве. Во второй строке находятся N целых чисел, по модулю не превосходящих 10^9 .

Формат выходного файла

В выходной файл надо вывести этот же массив в порядке неубывания, между любыми двумя числами должен стоять ровно один пробел.

Примеры

<code>sort.in</code>	<code>sort.out</code>
10	1 1 2 2 3 3 4 6 7 8
1 8 2 1 4 7 3 2 3 6	

Note

В этой задаче обязательно использовать быструю сортировку.

Задача D. Количество различных элементов в массиве

Имя входного файла: `differ.in`
Имя выходного файла: `differ.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дан массив целых чисел. Определите, сколько в нем содержится различных элементов.

Формат входного файла

Первая строка входного файла содержит целое число n ($1 \leq n \leq 10^5$). Вторая строка содержит n целых чисел, по модулю не превосходящих $2 \cdot 10^9$.

Формат выходного файла

Выведите единственное число — количество различных элементов в данном массиве.

Примеры

<code>differ.in</code>	<code>differ.out</code>
5 1 0 1 2 0	3
5 1 5 2 3 4	5
5 0 0 0 0 0	1