

1. Оптимизация сортировки пузырьком.

Реализация сортировки пузырьком, приведенная в лекции работает за N^2 при любых входных данных. Однако массив мог быть отсортирован на половине пути и все последующие итерации будут проходить впустую. Вам предлагается модифицировать данную реализацию, чтобы итерации проходили ровно до момента, когда массив станет отсортированным и после этого он заканчивал работу.

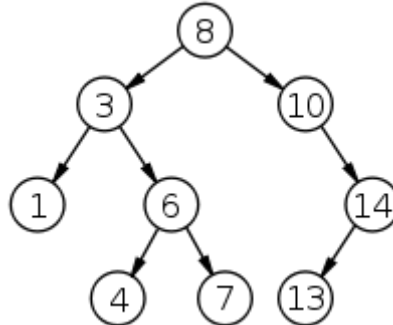
2. Сортировка деревом.

Алгоритм сортировки бинарным деревом немного похож на быструю сортировку. Основная идея заключается в следующем: есть бинарное дерево, в вершинах которого лежат числа. Изначально дерево состоит из одной вершины, в котором лежит какой-то элемент исходного массива (для определенности будем считать, что первый). Далее, в дерево будут добавляться все значения исходного массива следующим образом:

- Вначале это число подается вершине дерева.
- Если число меньше чем значение в этой вершине, то число будет передано в правое поддерево, если же больше, то в левое. Таким образом, все числа, которые будут лежать в правом поддереве будут больше чем это число, а все числа, которые будут лежать в левом поддереве будут меньше чем это число.
- Если подходящего поддерева нет, то создается новая вершина, в которую кладется число.
- После построения данного дерева достаточно обойти все вершины справа налево, чтобы восстановить отсортированный массив.

Вам предлагается реализовать данный алгоритм и подумать о том, насколько эффективно он будет работать.

Пример построения дерева для 8, 3, 1, 10, 14, 6, 4, 13, 7:



Далее если для каждой вершины начиная с корня выполнить процедуру

- Напиши значение в левом поддереве
- Напиши свое значение
- Напиши значение в правом поддереве

То результатом будет 1, 3, 4, 6, 7, 8, 10, 14, 13