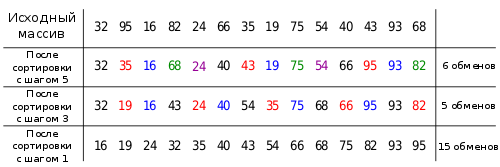
1. Сортировка пузырьком – это сортировка смысл которой заключается в том что она сравнивает 2 рядом стоящих элемента и переставляет их, идет с начала до конца, до тех пор пока есть хоть какие то перестановки.
2. Сортировка вставками – сортировка, поднимает элемент и если предыдущий элемент больше, то он сдвигается в право (при этом элемент который мы сравниваем запоминается), и это до тех пор пока не встретим элемент меньше того что мы передвигаем.
3. Сортировка выбором –сортировка, которая ищет наименьший элемент в не отсортированной части. Потом этот элемент вставляется в отсортированную часть а элемент который там стоял , ставят на место минимального.
4. Шейкерная сортировка – сортировка, пузырьковая сортировка которая движется направо, а затем налево, когда движемся направо несем туда самое большое число, налево самое маленькое, при этом блокируем их и сортируем элементы которые не заблокированы.
5. Сортировка Шелла – Пример:

Пусть дан список A = (32, 95, 16, 82, 24, 66, 35, 19, 75, 54, 40, 43, 93, 68) и выполняется его сортировка методом Шелла, а в качестве значений d выбраны 5, 3, 1.

На первом шаге сортируются подсписки A, составленные из всех элементов A, различающихся на 5 позиций, то есть подсписки A_{5,1} = (32, 66, 40), A_{5, 2} = (95, 35, 43), A_{5, 3} = (16, 19, 93), A_{5, 4} = (82, 75, 68), A_{5, 5} = (24, 54).

В полученном списке на втором шаге вновь сортируются подсписки из отстоящих на 3 позиции элементов.



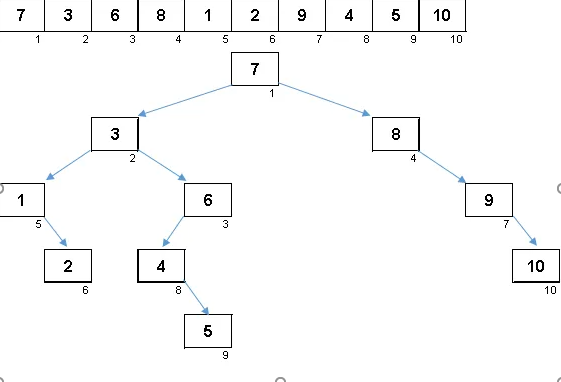
1. Алгоритм бинарного поиска – поиск какого либо числа в массиве, которая смотрит центральный элемент равен ли он искомому, если нет смотри больше он его или меньше, в зависимости от этого берет левую или правую часть массива и так до тех пор пока не найдет его.
2. Быстрая сортировка – сортировка, принцип

Выбрать из массива элемент, называемый опорным. Это может быть любой из элементов массива. (Последний)

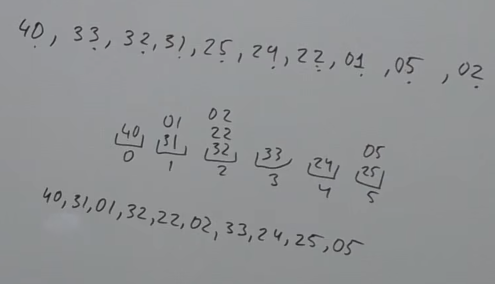
Сравнить все остальные элементы с опорным и переставить их в массиве так, чтобы разбить массив на три непрерывных отрезка, следующих друг за другом: «элементы меньшие опорного», «равные» и «большие».

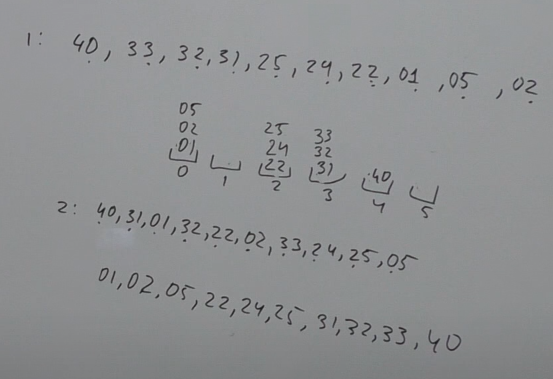
Для отрезков «меньших» и «больших» значений выполнить рекурсивно ту же последовательность операций, если длина отрезка больше единицы.

1. Внешняя сортировка слиянием – сортировка которая сохраняет элементы в файл.
2. Сортировка с помощью двоичного дерева – сортировка, есть дерево, берем 1 как корень и сортируем так что если следующий элемент меньше то сравниваем со следующим элементом, больше или равно справа.



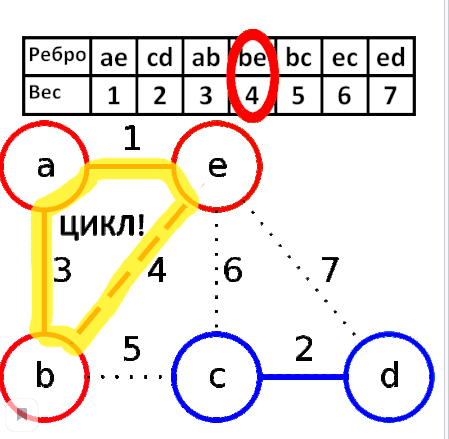
1. Поразрядная сортировка – сортировка, в которой смотрим максимальное кол-во разрядов в числе , у тех что меньше дописываем 0 слева, далее делаем проходов равное кол-ву разрядов. Сортируем массив по ячейкам у которых разряд равен разряду ячейки (слева на право по массиву, снизу в верх в ячейке ) выписываем обратно (с левой ячейки по правую, снизу в верх по ячейки), затем сортируем по следующему разряду. Выбор разряда, справа на лево



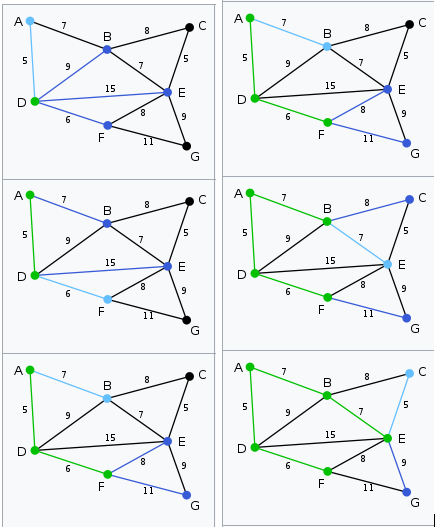


1. Хэш-таблицы с разрешением коллизий методом цепочек – таблица хешей в ячейке которой хранить лист коллизий.
2. Хэш-таблицы с разрешением коллизий методом открытой адресации – таблица хешей в ячейке элемент, а коллизия перепрыгивает на другую ячейку. (бывают разные виды)
3. ABC-сортировка – (алгортим сортировки слов)
4. Реализовать стек и базовые операции работы со стеком, с использованием собственного двусвязного списка – стек, стопка тарелок
5. Реализовать очередь и базовые операции работы с очередью, с использованием собственного двусвязного списка – очередь, первый пришел последним вышел
6. Алгоритм Крускала – в начале нужно отсортировать ребра по весу, затем поиск минимального остовного дерева, то есть ребра которые не образуют цикл

(Если нашли цикл пропустить это ребро)



1. Алгоритм Прима - Сначала берётся произвольная вершина затем от этой вершины ребро с наименьшей стоимостью, далее ребро из этих 2 вершин (вершины которые мы посетили не берем) и тд. До тех пор пока не соединим все вершины.



1. Обход графа в глубину - обход графа через стек
2. Обход графа в ширину - обход графа через очередь
3. Алгоритм Дейкстры – поиск кратчайшего пути из точки в точку, через прыжки то есть берем точку и делаем 1 прыжок то есть вершины которые соединены с этой вершиной будут иметь этот вес что и на ребре, но с 2 прыжками будем прыгать уже даже в другие , но если мы прыгнем в туже вершину в которой были и путь дешевле то перезаписать. При этом при каждом прыжке блокируем ту вершину в которой путь самый дешевый.

