Модуль 5

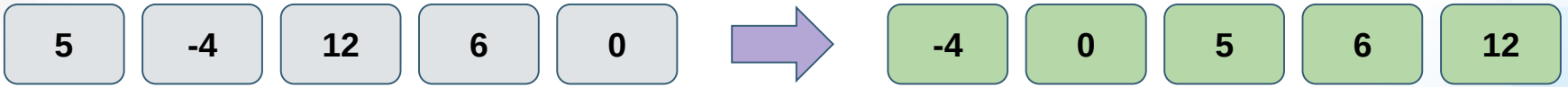
Алгоритмы сортировки и поиска

План модуля:

- Понятие сортировки
- Алгоритм сортировки выбором
- Алгоритм сортировки пузырьком
- Алгоритм быстрой сортировки
- Понятие сложности алгоритма
- Линейный поиск
- Бинарный поиск

Алгоритмы сортировки

Обсудим



? Зачем нужно сортировать списки?

? Предложите свой алгоритм сортировки

Алгоритмы сортировки

Сортировка выбором

5	-4	12	6	0

1. Находим минимальный элемент

5	-4	12	6	0

Алгоритмы сортировки

Сортировка выбором

5	-4	12	6	0

1. Находим минимальный элемент

5	-4	12	6	0

1. Записываем его в начало списка

-4	5	12	6	0

Алгоритмы сортировки

Сортировка выбором

5	-4	12	6	0

1. Находим минимальный элемент

5	-4	12	6	0

1. Записываем его в начало списка

-4	5	12	6	0

1. Снова ищем минимальный, в оставшейся части списка

-4	5	12	6	0

Алгоритмы сортировки

Сортировка выбором

5	-4	12	6	0

1. Находим минимальный элемент

5	-4	12	6	0

1. Записываем его в начало списка

-4	5	12	6	0

1. Снова ищем минимальный, в оставшейся части списка

-4	5	12	6	0

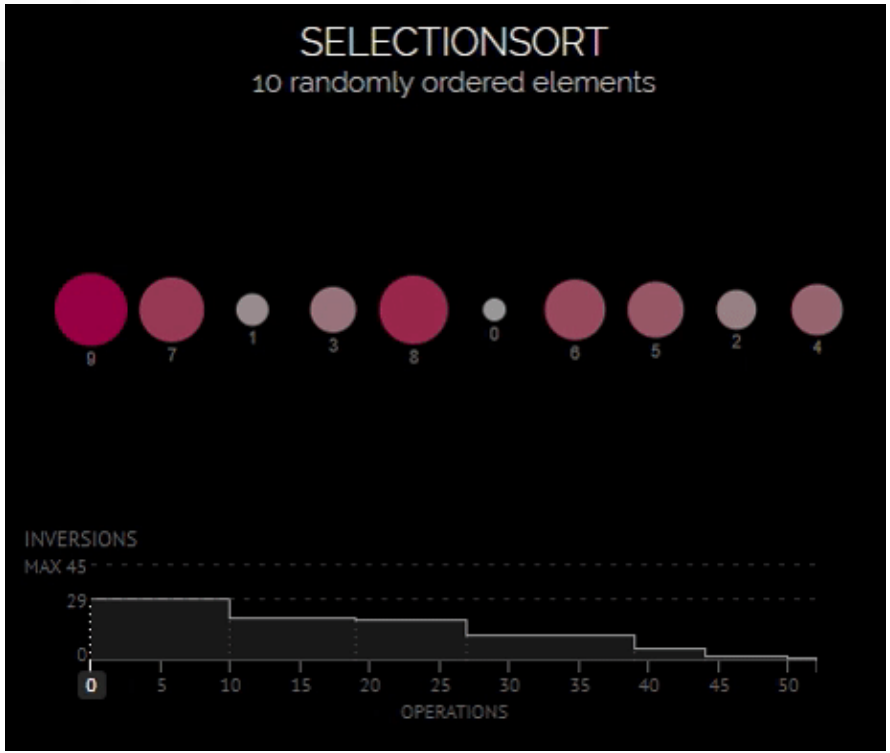
1. Второй минимум поместить на второе место списка

-4	0	12	6	5

1. Продолжать выполнять поиск и обмен, пока не будет достигнут конец списка

Алгоритмы сортировки

Сортировка выбором - визуализация



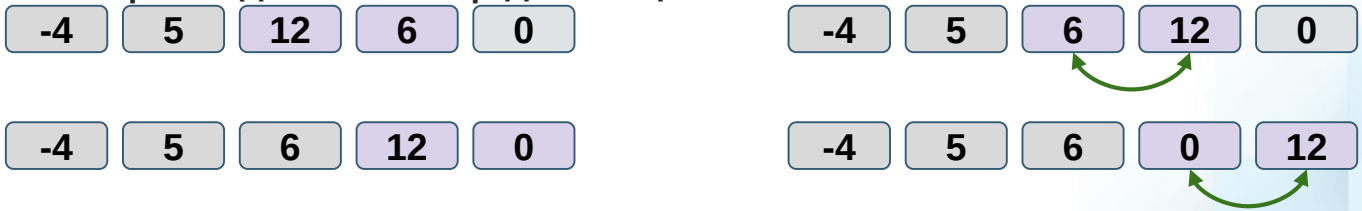
Алгоритмы сортировки

Сортировка “пузырьком”

1. Сравниваем два первых элемента, если первый больше, то меняем местами

1. Переходим к следующей паре, сравниваем, если нужно - меняем

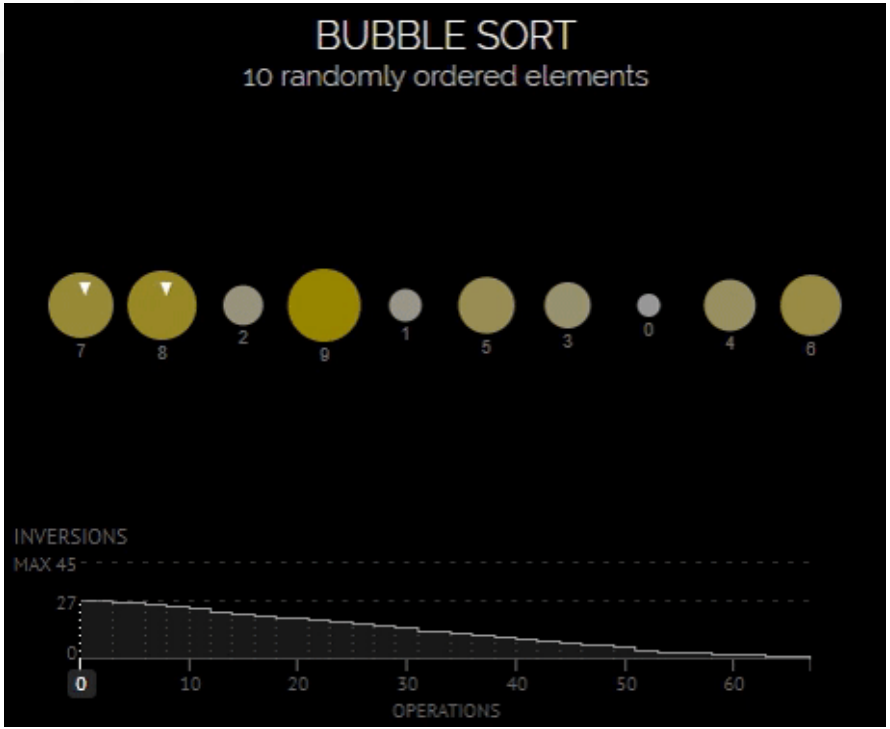
1. Повторяем для всех пар до конца



4. Повторяем алгоритм, без учета последнего

Алгоритмы сортировки

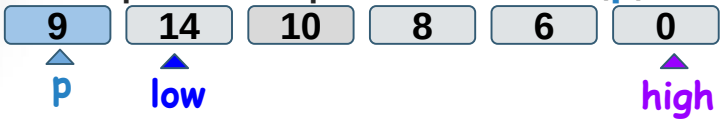
Сортировка “пузырьком”



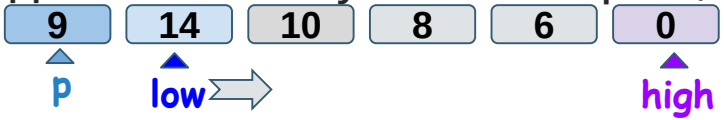
Алгоритмы сортировки

Сортировка quickSort

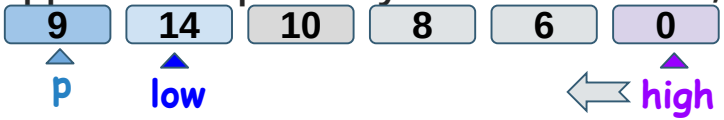
1. Выбираем опорный элемент(p) и введем два указателя

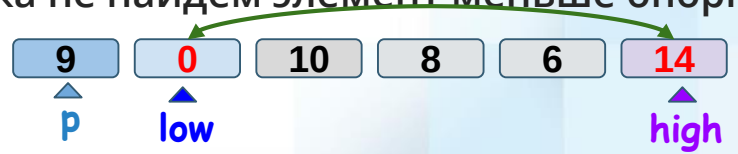


1. Двигаем левый указатель вправо, пока не найдем элемент больше(p)



1. Двигаем правый указатель влево, пока не найдем элемент меньше опорного(p)



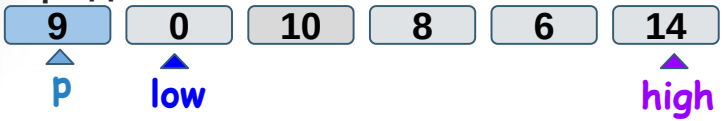


1. Если low < high, меняем элементы местами

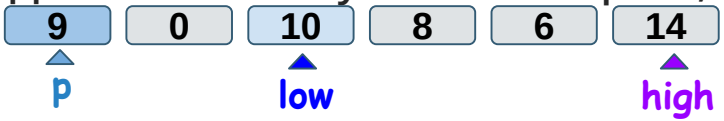
Алгоритмы сортировки

Сортировка quickSort

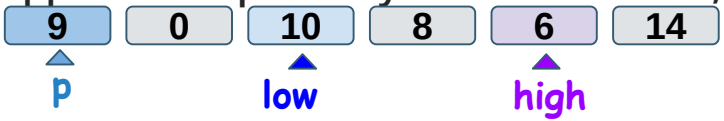
1. Продолжаем...



1. Двигаем левый указатель вправо, пока не найдем элемент больше(p)



1. Двигаем правый указатель влево, пока не найдем элемент меньше опорного(p)



1. Если low < high, меняем элементы местами

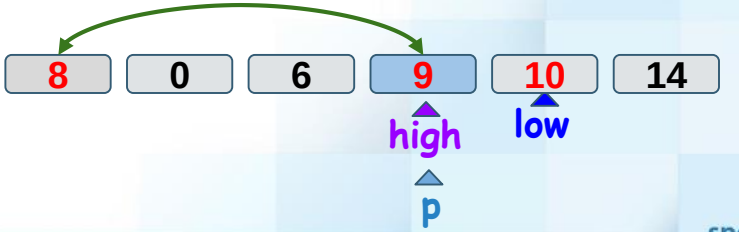
Алгоритмы сортировки

Сортировка quickSort

1. Двигаем указатели low и high далее, по тем же правилам.

1. Если low >= high, то разделение закончено

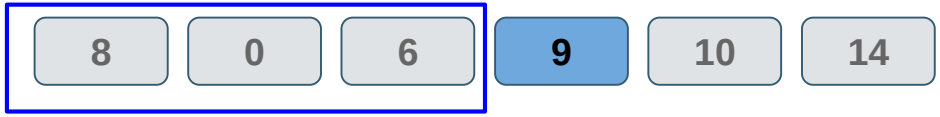
1. Двигаем правый указатель влево, пока не найдем элемент меньше опорного(p)



1. Меняем элементы местами high и p

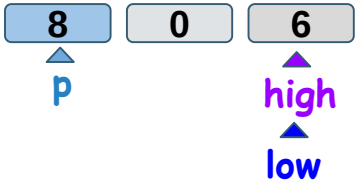
Алгоритмы сортировки

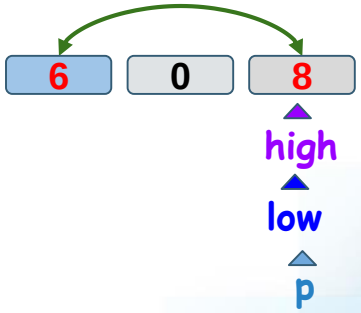
Сортировка quickSort



1. Двигаем указатели low и high далее, по тем же правилам.

1. Если low >= high, то разделение закончено

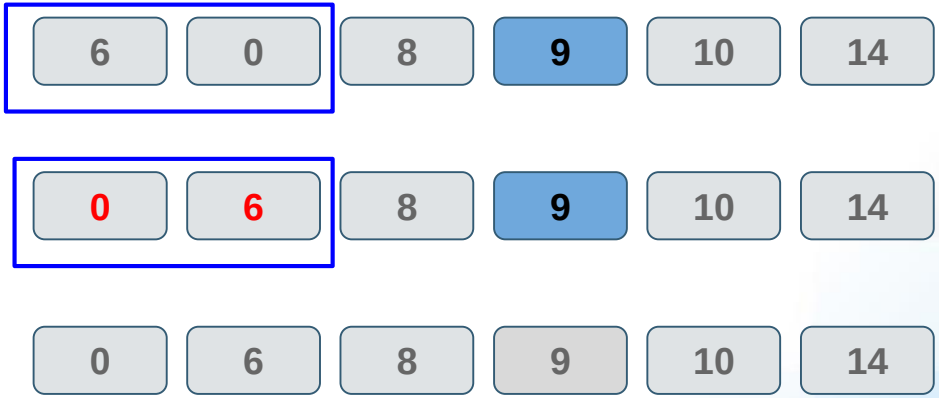




1. Меняем элементы местами high и p

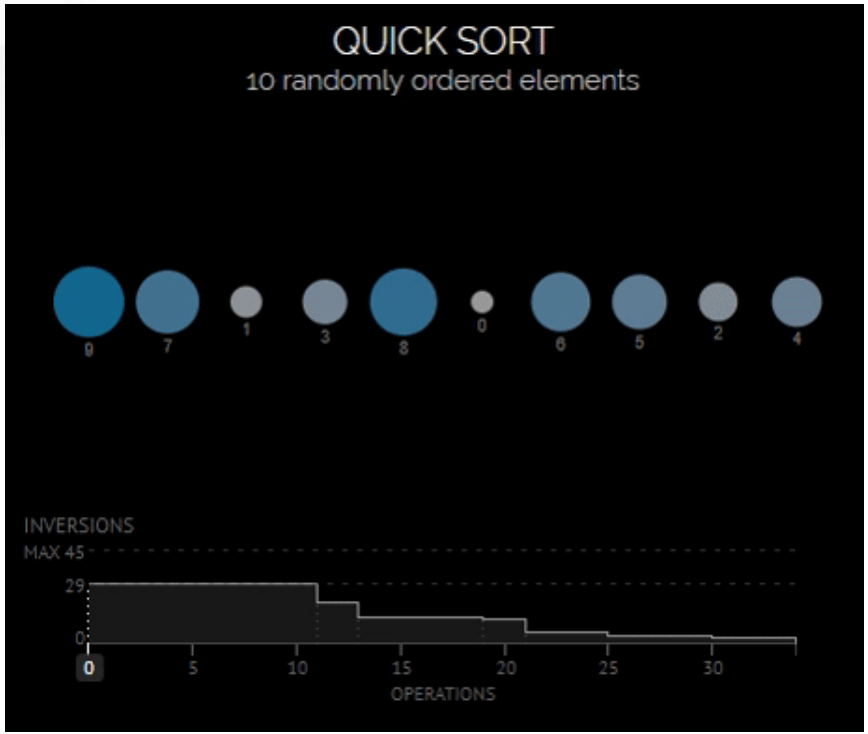
Алгоритмы сортировки

Сортировка quickSort



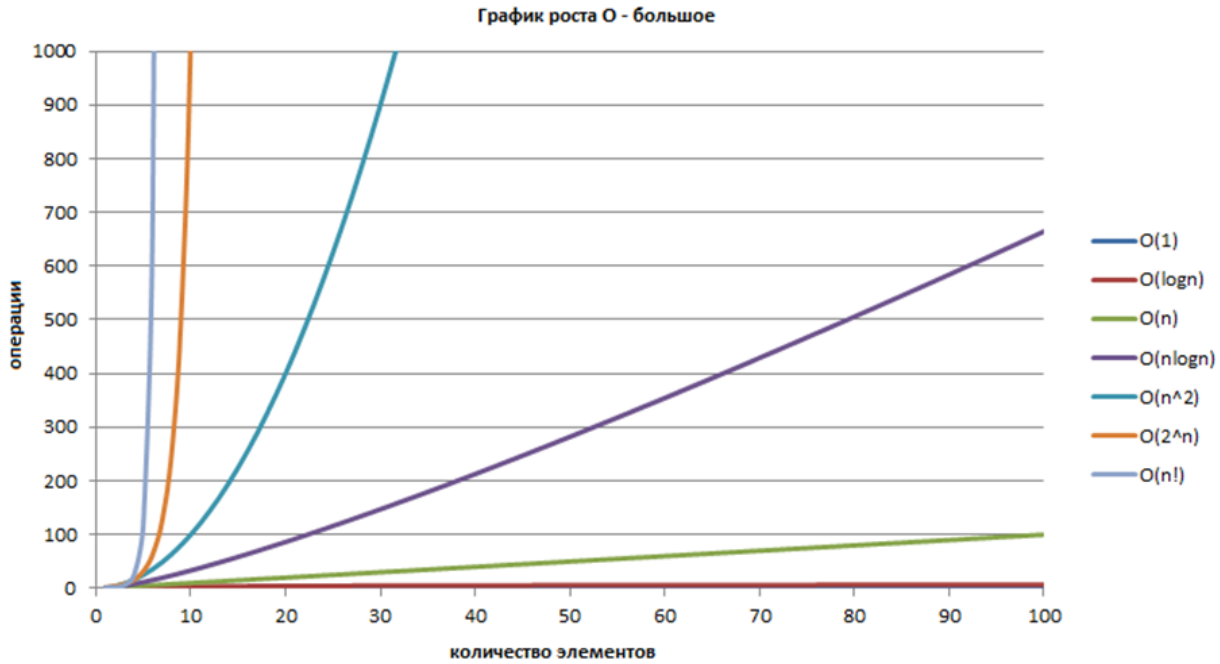
Алгоритмы сортировки

Быстрая сортировка



Сложность алгоритмов

Рост сложности алгоритма



Сложность алгоритмов сортировки

Таблица сложности алгоритмов сортировки

Алгоритм	Структура данных	Временная сложность			Вспомогательные данные
		Лучшее	В среднем	В худшем	В худшем
Быстрая сортировка	Массив	O(n log(n))	O(n log(n))	O(n^2)	O(n)
Сортировка слиянием	Массив	O(n log(n))	O(n log(n))	O(n log(n))	O(n)
Пирамидальная сортировка	Массив	O(n log(n))	O(n log(n))	O(n log(n))	O(1)
Пузырьковая сортировка	Массив	O(n)	O(n^2)	O(n^2)	O(1)
Сортировка вставками	Массив	O(n)	O(n^2)	O(n^2)	O(1)
Сортировка выбором	Массив	O(n^2)	O(n^2)	O(n^2)	O(1)
Блочная сортировка	Массив	O(n+k)	O(n+k)	O(n^2)	O(nk)
Поразрядная сортировка	Массив	O(nk)	O(nk)	O(nk)	O(n+k)