

Квадратичные сортировки

Выбором (D)

Есть список. Хотим отсортировать по неубыванию ↗ $a = [2\ 3\ 5\ 1\ 4]$

Алгоритм заключается в том, что ищем минимум и ставим его на 1-е место $\min = 1$

$a = [2\ 3\ 5\ 1\ 4]$ $a = [1\ 3\ 5\ 2\ 4]$ $a = [1\ 3\ 5\ 2\ 4]$ $a = [1\ 2\ 5\ 3\ 4]$ $a = [1\ 2\ 5\ 3\ 4]$ $a = [1\ 2\ 3\ 5\ 4]$ $a = [1\ 2\ 3\ 4\ 5]$	$a = [2\ 3\ 5\ 1\ 4]$ $a = [1\ 3\ 5\ 2\ 4]$ $a = [1\ 3\ 5\ 2\ 4]$ $a = [1\ 2\ 5\ 3\ 4]$ $a = [1\ 2\ 5\ 3\ 4]$ $a = [1\ 2\ 3\ 5\ 4]$ $a = [1\ 2\ 3\ 5\ 4]$ $a = [1\ 2\ 3\ 4\ 5]$
---	--

Оценим сложность:

Поиск первого минимума. $n-1$ сравнение. Следующего минимума $n-2$ сравнение. И т.д.

$n(n-1)/2$ сложность $O(n^2)$ такие сортировки называются квадратичными. Меняем найденный \min местами с текущим элементом 1 раз.

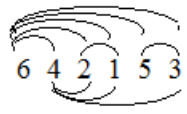
Пузырек (E)

Есть список. Хотим отсортировать по неубыванию ↗ $a = [6\ 4\ 2\ 1\ 5\ 3]$

Надо написать функцию. Идея: сравниваем 2 соседних элемента, если стоят неверно, то меняем их местами.

Подумать, почему в любой последовательности таких перестановок = количеству инверсий.

Инверсия 6 4 2 1 5 3

	$i < j$ и $a[i] > a[j]$
---	-------------------------

Одна перестановка уменьшает количество инверсий на 1

Max количество инверсий: $n(n-1)/2$

Min количество инверсий: 0

В пузырьке, за 1 проход, \max встанет на последнее место.

$a = [6\ 4\ 2\ 1\ 5\ 3]$

$a = [4\ 2\ 1\ 5\ 3\ 6]$

$a = [2\ 1\ 4\ 3\ 5\ 6]$

$a = [1\ 2\ 3\ 4\ 5\ 6]$

Итак, здесь 2 особенности:

1. Правая граница уменьшается.
2. Надо использовать Флаг, если ничего не переставили, то закончить сортировать.

Сложность алгоритма = количеству инверсий. В общем случае $O(n^2)$: $n(n-1)/2$

Для: $n\ 1\ 2\ \dots\ n-1$ сложность $O(n)$

Для: $2\ 3\ \dots\ 1\ n-1\ n$ сложность $O(n^2)$

Из квадратичных сортировок, лучшая вставкой.

Вставкой (F)

$a = [2\ 7\ 4\ 1\ 3\ 5]$

Левая часть отсортирована, просматриваем ее справа.

Рассмотрим префикс уже отсортированного **Вставкой** частично списка.

[2 7 ...] 4 надо двигать влево.

[2 4 7...] 1 надо двигать влево.

[1 2 4 7...] 3 надо двигать влево.

[1 2 3 4 7...] 5 надо двигать влево.

[1 2 3 4 5 7] Сложность в общем случае $O(n^2)$, до середины от половины.

Т.к. каждое передвижение, это одна инверсия. В среднем, инверсий: $n^2/4$.

Итак, ключевая идея: Нужную позицию искать справа, а не слева. Если искать слева, то это всегда $O(n^2)$