

Постановка задачи сортировки. Квадратичные сортировки

ceagest

1 Постановка задачи

Сортировка

1. Имеется последовательность из n элементов x_1, \dots, x_n .
2. Необходимо упорядочить элементы по неубыванию (невозрастанию). Для удобства далее будем везде сортировать по неубыванию.
3. Значит нужно найти перестановку индексов i_1, i_2, \dots, i_n такую, что $x_{i_1} \leq x_{i_2} \leq \dots \leq x_{i_n}$.

Определение 1.1. Сортировку называют основанной на сравнениях, если она работает в следующем предположении: объекты можно только сравнивать.

Определение 1.2. Устойчивость (стабильность) сортировки говорит о том, что элементы, одинаковые для сравнения, не меняют своего расположения относительно друг друга после сортировки.

2 Квадратичные сортировки

2.1 Сортировка пузырьком

- Выполним n итераций, в результате каждой из которых самый большой элемент «всплывает» в конец.
- В процессе итерации пробегаем из начала в конец и меняем соседние элементы, которые стоят не по порядку.
- Возможны такие оптимизации: «Если в процессе текущей итерации не было обменов – сортировка завершена», «можно идти на каждой итерации не до конца, а до $n - i$ »

2.2 Сортировка выбором

- На i -ой итерации ищем на отрезке $[i, n - 1]$ наименьший элемент и меняем его с i -ым местами.
- Таким образом за n итераций массив будет отсортирован.
- Если научиться как-то находить и извлекать минимум быстрее, чем за линейное время, можно получить очень хороший алгоритм.

2.3 Сортировка вставками

- На i -ой итерации ищем, куда вставить элемент на отрезке $[0, i]$. Таким образом, отрезок $[0, i]$ всегда отсортирован.
- Так за n итераций массив будет полностью отсортирован.

Замечание 2.1. Время работы всех трёх вышеперечисленных сортировок составляет $\mathcal{O}(n^2)$, притом для каждой из них требуется $\mathcal{O}(1)$ памяти. Более того, все эти сортировки являются стабильными.