



НИУ ВШЭ - Нижний Новгород

November 15, 2024

Алгоритмы и структуры данных

Лекция 2. Простые сортировки

Илья Сергеевич Бычков

ibychkov@hse.ru



Лекция 2.

Простые сортировки



Простые сортировки

План лекции

- 0. План лекции
- 1. Задача сортировки
- 2. Простые квадратичные сортировки
- 3. Специальные сортировки



Задача сортировки

Задача сортировки

Задача сортировки — это одна из основных задач в области алгоритмов и структур данных, которая заключается в упорядочивании набора объектов в соответствии с заданным критерием (оператором).

Примеры задачи сортировки:

- Сортировка списка учеников в алфавитном порядке по фамилии
- Сортировка товаров в интернет-магазине по цене от низкой к высокой
- Сортировка строк в таблице базы данных по дате создания
- Сортировка чисел в массиве по возрастанию
- Сортировка сотрудников компании по их должностям от высших к низшим



Простые сортировки

План лекции

0. План лекции
1. Задача сортировки
2. Простые квадратичные сортировки
3. Специальные сортировки



Простые квадратичные сортировки имеют следующие общие черты:

- основываются только на простом и последовательном сравнении элементов
- последовательно увеличивают текущую отсортированную часть массива добавляя один элемент
- имеют временную сложность $\mathcal{O}(n^2)$ в худшем случае

Простые сортировки:

- Сортировка выбором (**selection sort**)
- Сортировка вставками (**insertion sort**)
- Сортировка пузырьком (**bubble sort**)



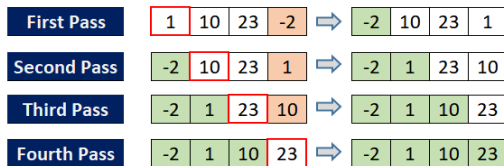
Простые квадратичные сортировки

Сортировка выбором

Сортировка выбором (selection sort)

Идея:

Последовательное определение i -го элемента отсортированного массива путем поиска минимума/максимума в оставшейся части (начиная с i).



Sorted Array



Smallest number in unsorted array



Correct position for smallest number in unsorted array



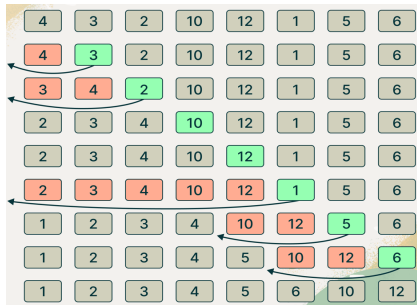
Простые квадратичные сортировки

Сортировка вставками

Сортировка вставками (insertion sort)

Идея:

Поиск корректной позиции рассматриваемого элемента в уже отсортированной части массива.



(2) Сортировка вставками, Источник - [Dev.to](#)



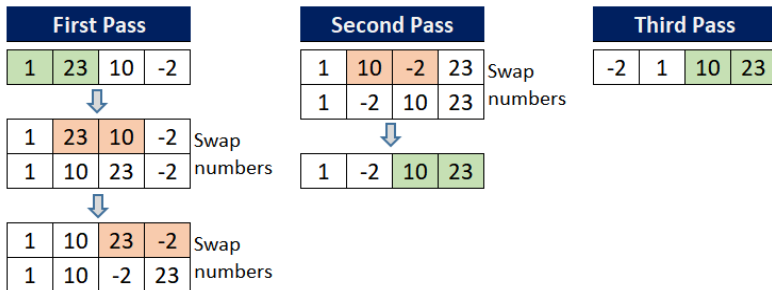
Простые квадратичные сортировки

Сортировка пузырьком

Сортировка пузырьком (bubble sort)

Идея:

Попарное сравнение соседних элементов, которое позволяет текущему "лучшему" элементу "всплыть"





Простые сортировки

План лекции

- 0. План лекции
- 1. Задача сортировки
- 2. Простые квадратичные сортировки
- 3. Специальные сортировки



Специальные сортировки

Сортировка ?

Задача:

Дан массив, состоящий из 1000 целых чисел, каждое из которых находится в интервале $[0, 100]$.

Отсортируйте данный массив.



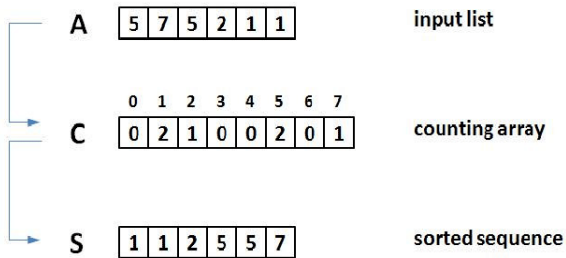
Специальные сортировки

Сортировка подсчётом

Задача:

Дан массив, состоящий из 1000 целых чисел, каждое из которых находится в интервале $[0, 100]$. Отсортируйте данный массив.

Ответ: Сортировка подсчётом (counting sort)



(4) Сортировка подсчётом, Источник - -



Блочная/карманная сортировка (bucket sort)

Для карманной сортировки нужно разбить элементы массива входных данных на k блоков (карманов, корзин). Далее каждый из таких блоков сортируется либо другой сортировкой, либо рекурсивно тем же методом разбиения. После сортировок внутри каждого блока данные записываются в массив в порядке разбиения на блоки.

Сложность:

Worst-case time complexity: $\mathcal{O}(n^2)$

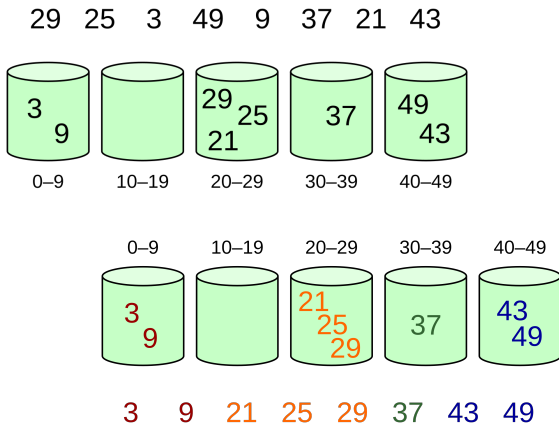
Worst-case space complexity: $\mathcal{O}(n \cdot k)$



Специальные сортировки

Блочная/карманная сортировка

Блочная/карманная сортировка (bucket sort)



(6) Сортировка подсчётом, Источник - [Wikipedia](#)



Специальные сортировки

Блочная/карманная сортировка

Блочная/карманная сортировка (bucket sort)

```
double[] bucketSort(double[] array)
    minElement = Infinum
    maxElement = -Infinum
    for i = 0 to array.length - 1
        minElement = minimum(minElement, array[i])
        maxElement = maximum(maxElement, array[i])
    range = maxElement - minElement
    for i = 0 to array.length - 1
        index = int(array[i] * numBuckets / range)
        добавим array[i] в конец buckets[index]
    for i = 0 to numBuckets - 1
        buckets[i] = sort(buckets[i])
    for i = 0 to numBuckets - 1
        for k = 0 to buckets[i].length - 1
            добавим buckets[i][k] в конец answer
    return answer
```

(7) Сортировка подсчётом, Источник - [neerc.ifmo](http://neerc.ifmo.ru)

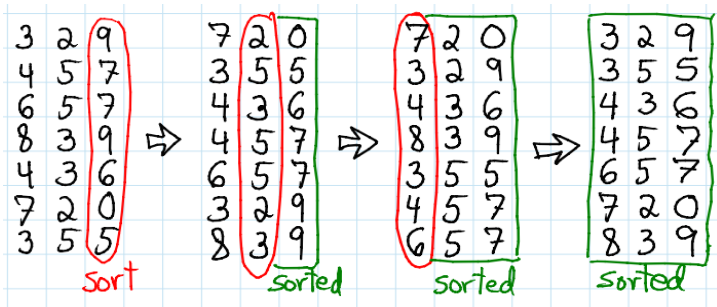


Специальные сортировки

Поразрядная сортировка

Поразрядная сортировка (radix sort)

Разделение элементов массива на группы по их цифрам (разрядам) в определённой позиции и сортировка таких групп. Этот процесс повторяется для каждой позиции (разряда) числа, пока все элементы не будут упорядочены.





Специальные сортировки

Поразрядная сортировка

Поразрядная сортировка (radix sort)

Сортировка по младшему разряду (для чисел)

| | | | | | | | | | |
|-----|---|----|----|---|----|----|-----|----|-----|
| 101 | 1 | 20 | 50 | 9 | 98 | 27 | 153 | 35 | 899 |
|-----|---|----|----|---|----|----|-----|----|-----|



| | | | | | | | | | |
|----|-----|---|-----|---|----|---|----|----|-----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 20 | 101 | | 153 | | 35 | | 27 | 98 | 9 |
| 50 | 1 | | | | | | | | 899 |



| | | | | | | | | | |
|----|----|-----|---|-----|----|----|----|---|-----|
| 20 | 50 | 101 | 1 | 153 | 35 | 27 | 98 | 9 | 899 |
|----|----|-----|---|-----|----|----|----|---|-----|

(9) Поразрядная сортировка, Источник - [alphacodingskills](#)



Специальные сортировки

Поразрядная сортировка

Поразрядная сортировка (radix sort)

Сортировка по разряду десятков

| | | | | | | | | | |
|----|----|-----|---|-----|----|----|----|---|-----|
| 20 | 50 | 101 | 1 | 153 | 35 | 27 | 98 | 9 | 899 |
|----|----|-----|---|-----|----|----|----|---|-----|



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|---|----|----|---|-----|---|---|---|-----|
| 101 | | 20 | 35 | | 50 | | | | 98 |
| 01 | | 27 | | | 153 | | | | 899 |
| 09 | | | | | | | | | |



| | | | | | | | | | |
|-----|---|---|----|----|----|----|-----|----|-----|
| 101 | 1 | 9 | 20 | 27 | 35 | 50 | 153 | 98 | 899 |
|-----|---|---|----|----|----|----|-----|----|-----|

(10) Поразрядная сортировка, Источник - [alphacodingskills](#)



Специальные сортировки

Поразрядная сортировка

Поразрядная сортировка (radix sort)

Сортировка по старшему разряду

| | | | | | | | | | |
|----|----|-----|---|-----|----|----|----|---|-----|
| 20 | 50 | 101 | 1 | 153 | 35 | 27 | 98 | 9 | 899 |
|----|----|-----|---|-----|----|----|----|---|-----|



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|-----|---|---|---|---|---|---|-----|---|
| 001 | 101 | | | | | | | 899 | |
| 009 | 153 | | | | | | | | |
| 020 | | | | | | | | | |
| 027 | | | | | | | | | |
| 035 | | | | | | | | | |
| 050 | | | | | | | | | |
| 098 | | | | | | | | | |



| | | | | | | | | | |
|---|---|----|----|----|----|----|-----|-----|-----|
| 1 | 9 | 20 | 27 | 35 | 50 | 98 | 101 | 153 | 899 |
|---|---|----|----|----|----|----|-----|-----|-----|

(11) Поразрядная сортировка, Источник - [al-phacodingskills](#)



Поразрядная сортировка (radix sort)

Существует два варианта поразрядной сортировки:

- **Least Significant Digit (LSD)** - от младшего разряда к старшему. Сначала идут более короткие элементы. Корректный порядок для чисел.

Пример: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

- **Most Significant Digit (MSD)** - от старшего разряда к младшему. Лексикографический порядок. Подходит для сортировки строк.

Пример: [1, 10, 2, 3, 4, 5, 6, 7, 8, 9]