

Простые сортировки

Алексей Владыкин

СПбГУ ИТМО

26 октября 2009

Определения

- Сортировка — упорядочение элементов массива в соответствии с каким-либо отношением линейного порядка.
- Устойчивая сортировка сохраняет относительный порядок эквивалентных элементов.
- Внутренняя сортировка: все данные влезают в оперативную память.
- Внешняя сортировка: данные не влезают в память и хранятся на внешних устройствах.

Беспорядок (инверсия)

- Беспорядок: $i < j \ \&\& \ a[i] > a[j]$.
- В отсортированном по возрастанию массиве 0 беспорядков.
- Сколько беспорядков в отсортированном по убыванию массиве?

Беспорядок (инверсия)

- Беспорядок: $i < j \ \&\& \ a[i] > a[j]$.
- В отсортированном по возрастанию массиве 0 беспорядков.
- Сколько беспорядков в отсортированном по убыванию массиве?

$$\frac{n(n-1)}{2}$$

Беспорядок (инверсия)

- Беспорядок: $i < j \ \&\& \ a[i] > a[j]$.
- В отсортированном по возрастанию массиве 0 беспорядков.
- Сколько беспорядков в отсортированном по убыванию массиве?

$$\frac{n(n-1)}{2}$$

- На сколько изменится количество беспорядков при обмене двух соседних элементов?

Беспорядок (инверсия)

- Беспорядок: $i < j \ \&\& \ a[i] > a[j]$.
- В отсортированном по возрастанию массиве 0 беспорядков.
- Сколько беспорядков в отсортированном по убыванию массиве?

$$\frac{n(n-1)}{2}$$

- На сколько изменится количество беспорядков при обмене двух соседних элементов?
- На сколько изменится количество беспорядков при обмене двух произвольных элементов?

Пузырьковая сортировка

На i -й итерации внешнего цикла «всплывает» i -й максимум.
Обмениваются только соседние элементы.

```
void bubble_sort(int a[], int n) {
    int i, j;
    for (i = n - 1; i > 0; --i) {
        for (j = 0; j < i; ++j) {
            if (a[j] > a[j + 1]) {
                swap(&a[j], &a[j + 1]);
            }
        }
    }
}
```

Оптимизация: шейкерная сортировка.

Ещё одна пузырьковая сортировка?

На i -й итерации внешнего цикла ставит на место i -й минимум.

```
void bubble_sort2(int a[], int n) {
    int i, j;
    for (i = 0; i < n - 1; ++i) {
        for (j = i; j < n; ++j) {
            if (a[j] < a[i]) {
                swap(&a[i], &a[j]);
            }
        }
    }
}
```


Сортировка выбором

На i -й итерации внешнего цикла ставит на место i -й максимум.

```
void selection_sort(int a[], int n) {
    int i, j, j_max;
    for (i = n - 1; i > 0; --i) {
        j_max = 0;
        for (j = 1; j <= i; ++j) {
            if (a[j_max] < a[j]) {
                j_max = j;
            }
        }
        swap(&a[j_max], &a[i]);
    }
}
```

Сортировка вставками

На i -й итерации внешнего цикла вставляет новый элемент в отсортированный подмассив длины i .

```
void insertion_sort(int a[], int n) {
    int i, j, t;
    for (i = 1; i < n; ++i) {
        t = a[i];
        for (j = i - 1; 0 <= j && t < a[j]; --j) {
            a[j + 1] = a[j];
        }
        a[j + 1] = t;
    }
}
```

Бинарные вставки. Сортировка Шелла.