

Сортировки: новолуние

Мы с вами уже изучили некоторые сортировки,
но у них есть некоторые недостатки

Они медленные! Можно быстрее!

Давайте рассмотрим такой сюжет

Давайте рассмотрим такой сюжет

- У вас есть два отсортированных массива x_1 и x_2

Давайте рассмотрим такой сюжет

- У вас есть два отсортированных массива x_1 и x_2
- Вы хотите получить новый отсортированный массив x

Давайте рассмотрим такой сюжет

- У вас есть два отсортированных массива x_1 и x_2
- Вы хотите получить новый отсортированный массив x
- Как это сделать за $O(n)$?

Давайте рассмотрим такой сюжет

- У вас есть два отсортированных массива x_1 и x_2
- Вы хотите получить новый отсортированный массив x
- Как это сделать за $O(n)$?
- (Давайте я это нарисую)

Сортировка слиянием

Сортировка слиянием

- Изначально имеется неотсортированный массив

Сортировка слиянием

- Изначально имеется неотсортированный массив
- Делим его на две части

Сортировка слиянием

- Изначально имеется неотсортированный массив
- Делим его на две части
- Сортируем каждую из частей (сейчас объясню как)

Сортировка слиянием

- Изначально имеется неотсортированный массив
- Делим его на две части
- Сортируем каждую из частей (сейчас объясню как)
- Сделаем слияние двух отсортированных массивов

Сортировка слиянием

- Изначально имеется неотсортированный массив
- Делим его на две части
- Сортируем каждую из частей (сейчас объясню как)
- Сделаем слияние двух отсортированных массивов
- Получим отсортированный массив

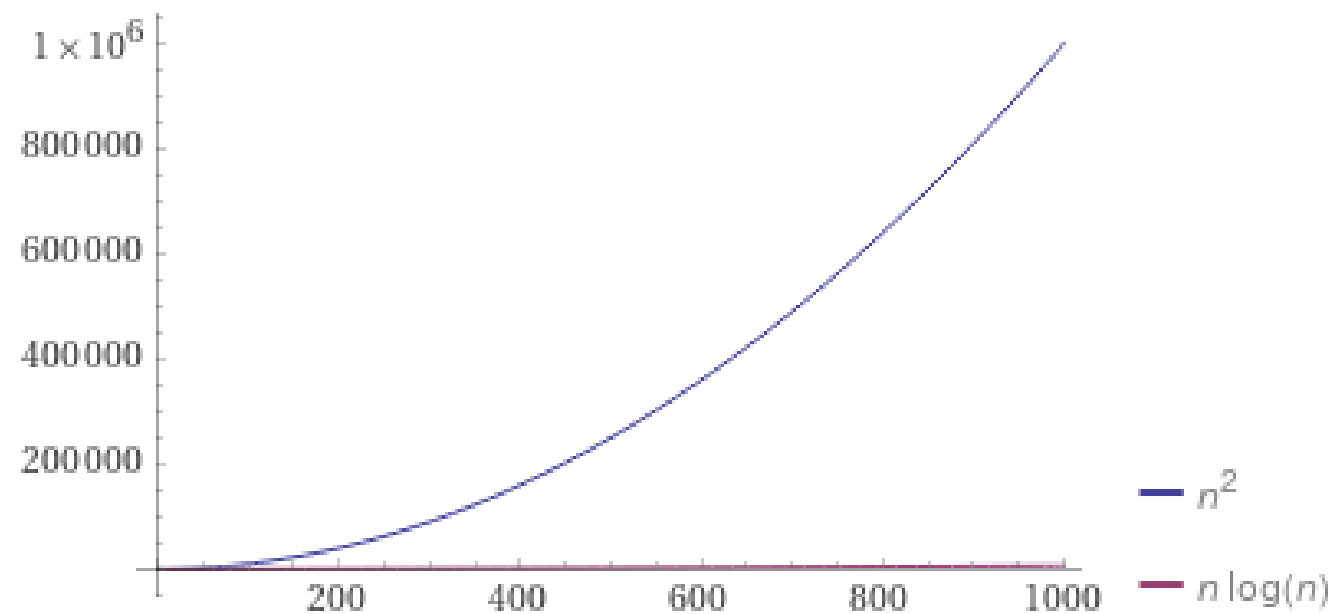
Как сортировать промежуточные массивы?

- А точно также
- Как тогда остановиться?
- Будем их тоже сортировать этой сортировкой!
- В какой-то момент у нас будет массив из нуля или одного элемента, а они уже являются отсортированными
- Звучит как рекурсия, давайте посмотрим на коде

Что с асимптотикой?

- Давайте порисуем

Поздравляю кого-то с первой сортировкой
за $n \log n$



Есть ещё одна сортировка...

Быстрая сортировка

Быстрая сортировка

- Берём элемент в массиве

Быстрая сортировка

- Берём элемент в массиве
- Делим массив на три группы элементов – меньше исходного, равны исходному, больше исходного

Быстрая сортировка

- Берём элемент в массиве
- Делим массив на три группы элементов – меньше исходного, равны исходному, больше исходного
- С первой и третьей группой элементов продолжаем сортировку

Псевдокод

```
void quicksort(a: T[n], int l, int r)
    if l < r
        int q = partition(a, l, r)
        quicksort(a, l, q)
        quicksort(a, q + 1, r)
```

```
int partition(a: T[n], int l, int r)
    T v = a[(l + r) / 2]
    int i = l
    int j = r
    while (i ≤ j)
        while (a[i] < v)
            i++
        while (a[j] > v)
            j--
        if (i ≥ j)
            break
        swap(a[i++], a[j--])
    return j
```

Асимптотика?

- Тут не так всё очевидно

Сейчас ещё покодим стандартную
сортировку