Сортировки

Сортировка вставками

Инвариант - поле i-го шага отсортирован префикс [0, i]

```
for i=0..n-1:
    j = i
    while j > 0 && a[j] < a[j - 1]:
        swap(a[j], a[j - 1])
        j--
```

Время работы: n^2

Сортировка слиянием (Merge Sort)

> Разделяй и властвуй!

Делим напоплоам, потом делаем merge? Далее - рекурсивно.

```
fn merge(l: Vec<i64>, r: Vec<i64>) -> Vec<i64> {
    let mut res = Vec::new();

    let mut i = 0 as usize;
    let mut j = i;

    while i + j < l.len() + r.len() {
        if i != l.len() && (j == r.len() || l[i] < r[j]) {
            res.push(l[i]);
            i += 1;
        }
        else {
            res.push(r[j]);
            j += 1;
        }
    }

    return res;
}

fn sorted(v: Vec<i64>) -> Vec<i64> {
    let n = v.len();
    let m = v / 2usize;

    let res = Vec::new();

}
```

Мастер теорема

Если

$$T(n) \leq b \times T\left(\frac{n}{a}\right) + n^c$$

$$T(n) = \begin{cases} n^{log_a b}, c < log_a b \\ n^c, c > log_a b \\ n^c \times \log n, c = \end{cases}$$