Способы оценки времени работы:

- Прямой учет
- Рекурсивная оценка
- Амортизационный анализ

Прямой учет Время работы строчки — произведение верхних оценок по всем строчкам-предкам нашей.

К примеру

```
while (!is_sorted()) { // O(# inversions) = O(n^2)
    for (int i = 0; i + 1 < n; i++) { // O(n) * O(parent) = O(n^3)
        if (a[i] > a[i + 1]) {
            swap(a[i], a[i + 1]); // O(n^3)
        }
    }
}
```

Рекурсивная оценка Пример — сортировка слиянием

Нас интересует две вещи: инвариант и переход. Для оценки времени используем рекурренту вида $T(n) = O(f(n)) + \sum\limits_{n' \in calls} T(n')$

При этом если мы доказываем время работы, то показываем $T(n) \le c \cdot f(n)$, зная, что для n' $\exists c : T(n') \le c \cdot f(n)$

Важно, что c глобальное и не должно увеличиваться в ходе доказательства

stable sort Делает сортировку, не меняя порядок равных элементов относительно исходной последовательности. Merge-sort стабилен.

inplace-algorithm Не требует дополнительной памяти и делает все прямо на данной памяти (у нас есть $\log n$ памяти на рекурсию). Quick-sort inplace.

Время работы qsort

$$T(n) = \max_{input} average_{rand} t(input, rand) = \max_{|input|=n} Et(input)$$

$$T(n) \le \Theta(n) + \frac{1}{n}\sum_{k=0}^{n-1}(T(k+1) + T(n-k)) \le \Theta(n) + \frac{2}{n} \cdot \sum_{k=1}^{n} T(k-1) \le a \cdot n + \frac{2}{n}\sum_{k=1}^{n} c \cdot (k-1) \cdot \log(k-1) \le$$

$$\le a \cdot n + \frac{2}{n}\sum_{k=1}^{\frac{n}{2}} c \cdot (k-1) \cdot \log n - \frac{2}{n}\sum_{k=1}^{\frac{n}{2}} c \cdot (k-1) + \frac{2}{n}\sum_{k=\frac{n}{2}+1}^{n} c \cdot (k-1) \cdot \log n \le$$

$$\le a \cdot n + \frac{2}{n} \cdot n^2 \cdot \log n - \frac{2c}{n} \cdot \frac{(n-2)^2}{4} \le a \cdot n + cn\log n - \frac{c(n-2)}{4} \le cn\log n$$

$$\frac{c(n-2)}{4} \geq a \cdot n$$

$$c \cdot n - 2c \geq 4 \cdot a \cdot n$$

$$c \geq \frac{4 \cdot a \cdot n}{(n-2)}$$

, что верно для достаточно больших n.

Ограничение на число сравнений в сортировке Бинарные сравнения на меньше.

Рассмотрим дерево переходов. Для перестановки есть хотя бы один лист — листьев котя бы $n!$

$$L(T) \leq 2^x, d(T) \leq x$$

, если x — ответ

$$L(T) \geq n!$$

$$d(T) \geq \log n! \geq \log \frac{n}{2}^{\frac{n}{2}} = \log 2^{(\log \frac{n}{2}) \cdot \frac{n}{2}} = \frac{n}{2} \cdot \log \frac{n}{2} = \Omega(n \log n)$$