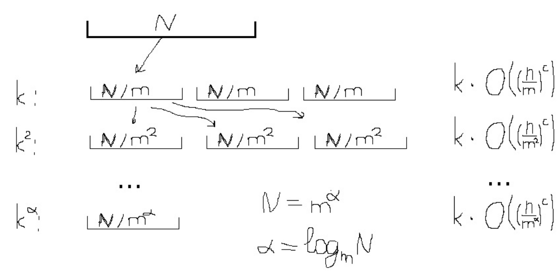
Master-theorem

$$T(n) = \begin{cases} O(1); n \le 1 \\ k * T(\frac{n}{m}) + O(n^c); n > 1 \end{cases}$$

1. $c > log_m k \to T(n) = O(n^c)$ – если подзадачи просты, но разбиение сложное
2. $c < log_m k \to T(n) = O(n^{log_m k})$ – если подзадачи сложны, но разбиение простое
3. $c = log_m k \to T(n) = O(n^c * n^{log_m k})$ – если задачи и разбиение эквивалентны по сложности

$$O(k^i * (\frac{n}{m^i})^c) = O(n^c * (\frac{k}{m^c})^i),$$

$$\frac{k}{m^c} <> 1.$$



1. $log_m k > c \to T(n) = \sum_{i=0}^{log_m n} O(n^c * (\frac{k}{m^c})^i) = O(n^c * \sum_{i=0}^{log_m n} (\frac{k}{m^c})^i) = O(n^c * (\frac{k}{m^c})^{log_m n}) = O(n^c * \frac{k^{log_m n}}{n^c}) = O(k^{log_m n}) = O(n^{log_m k})$
2. $log_m k < c \to T(n) = O(\sum_{i=0}^{log_m n}(n^c * (\frac{k}{m^c})^i)) = O(n^c * \sum_{i=0}^{log_m n}(\frac{k}{m^c})^i) = O(n^c)$
3. $log_m k = c \to T(n) = O(\sum_{i=0}^{log_m n}(n^c * (\frac{k}{m^c})^i)) = O(n^c * \sum_{i=0}^{log_m n}(\frac{k}{m^c})^i) = O(n^c * \sum_{i=0}^{log_m n} 1^i) = O(n^c * (log_m n + 1)) = n^c * O(log(n)) = O(n^c * log(n))$

Разделяй и властвуй в бинарном поиске:

$$m = 2, k = 1, c = 0 \to log_m k = log_2 1 = 0 = c \to T(n) = O(log(n))$$

Разделяй и влавствуй в mergesort:

$$m = 2, k = 2, c = 1 \to log_m k = log_2 2 = 1 = c \to T(n) = O(n * log(n))$$

Рассмотрим $f(n) = n * \sqrt{n+1} \le n\sqrt{2n} = O(n^{3/2})$

$$T(n) = \begin{cases} 2T(n/3) + O(f(n)), n > 1 \\ d, n \le 1 \end{cases}$$

$\to m = 3, k = 2, c = 3/2$

Алгоритм Карацубы

Рассмотрим умножение в столбик: $O(n)$ от перемножения строчки на константу. Суммируем n чисел длины n. Сложность умножения в столбик $O(n^2)$.
$A{*}B = (A_1{*}10^n{+}A_2)(B_1{*}10^n{+}B_2) = A_1B_1{*}10^{2n}{+}(A_1B_2{+}A_2B_1){*}10^n{+}B_1B_2$ – пока выигрыша нет.

$$(A_1 + A_2)(B_1 + B_2) = A_1B_1 + A_2B_2 + (A_1B_2 + A_2B_1),$$
$$A_1B_2 + A_2B_1 = (A_1 + A_2)(B_1 + B_2) - A_1B_1 - A_2B_2.$$

Сложность алгоритма: $O(n^{log_2 3})$. При этом существуют более совершенные алгоритмы умножения.

Быстрая сортировка

Возьмём X. Отделим наборы большие X и меньшие X. X встанет на своё место. Возьмём X1 и X2. Повторим операцию до конца.
Хороший случай: $O(n * log(n))$
Плохой случай: $O(n^2)$