

Математические основы алгоритмов

А. С. Охотин

Определение 1. *Машина с произвольным доступом в память (RAM).* У нас есть память в виде ячеек на \mathbb{Z} , где хранятся программы и следующие операции:

- присваивание: $A = B$;
- арифметические операции: $A = B + C$, $A = B - C$, $A = B \times C$, $A = B / C$, $A = B \div C$;
- GOTO n ;
- GOTO x_n ;
- IF $A == B$ THEN GOTO n (вместо $A == B$ может быть $A >= B$; вместо n может быть x_n);

Определение 2. *Сложность работы программы* — это такая функция $t(n)$ равная максимуму затрачиваемых ходов по всем входным данным длины n .

Замечание 1. Рассматривают ещё модель “log-cost”, где каждая операция стоит логарифм от входящих в неё значений.

Определение 3. *Сложность памяти программы* — это такая функция $s(n)$ равная максимуму затрачиваемого места по всем входным данным длины n .

Теорема 1 (Карацуба). *Умножение двух чисел длины не более n можно посчитать за время $O(n^{\log_2(3)})$.*

Теорема 2. *Давите разобьём наши числа $\overline{a_{n-1} \dots a_0}$ и $\overline{b_{n-1} \dots b_0}$ на две примерно равные половины: $\overline{a_{n-1} \dots a_0} = \overline{A_1 A_2}$, $\overline{b_{n-1} \dots b_0} = \overline{B_1 B_2}$. Тогда произведение (без переносов переполнения разрядов) равно*

$$\overline{A_1 B_1; (A_1 + A_2)(B_1 B_2) - A_1 B_1 - A_2 B_2; A_2 B_2}$$

Видно, что для этого нужно сделать три умножения и четыре сложения (вычитания). Т.е. $T(n) = 3T(n/2) + O(n)$. Тем самым $T(n) = O(n^{\log_2(3)})$.

Теорема 3. *Сортировка добавлением элемента работает за $\approx n^2/2$.*

Теорема 4. *Сортировка слиянием (merge sort) работает за $O(n \log(n))$.*