

Сложность некоторых алгоритмов

Сложность алгоритмов. Сложность некоторых алгоритмов



Рассмотрим примеры алгоритмов и посчитаем их вычислительную сложность.

Минимум в списке

Дан список из элементов a_0, a_1, \dots, a_{n-1} , необходимо найти в нём минимальный элемент.

Неважно, будем ли мы использовать для решения задачи встроенную в язык программирования функцию или напишем решение сами, количество операций данного алгоритма будет составлять $n - 1$, что соответствует асимптотике $O(n)$. То есть этот алгоритм является линейным от количества элементов списка.

Удаление последнего элемента

Дан список из элементов a_0, a_1, \dots, a_{n-1} , необходимо удалить в нём последний элемент.

В данной задаче все элементы, кроме последнего, остаются на своих местах, и вычислительная сложность алгоритма составляет $O(1)$.

Удаление произвольного элемента

Дан список из элементов a_0, a_1, \dots, a_{n-1} , необходимо удалить в нём элемент с индексом i .

При удалении элемента с индексом i все стоящие за ним элементы сдвигаются влево. В худшем случае (при удалении элемента с индексом 0) алгоритм может выполнить $n - 1$ операцию. Следовательно, вычислительная сложность такого алгоритма $O(n)$. Данный алгоритм является линейным.

Перебор пар элементов в списке

Дан список из элементов a_0, a_1, \dots, a_{n-1} , необходимо вывести все пары (a_i, a_j) такие, что $i < j$.

Количество таких пар составляет $(n - 1) + (n - 2) + \dots + 2 + 1 = \frac{n(n-1)}{2}$. Асимптотика алгоритма составляет $O(n^2)$, то есть алгоритм является квадратичным.

Перебор n -значных чисел в двоичной системе счисления

Необходимо вывести все n -значные числа в двоичной системе счисления (числа с ведущими нулями учитываются). Например, для $n = 3$ необходимо вывести 8 чисел:

000
001
010
011
100
101
110
111

Количество таких чисел составляет 2^n . Поэтому алгоритм относится к классу экспоненциальных, и его асимптотика составляет как минимум $O(2^n)$.

Проверка числа на простоту

Задано число n , необходимо проверить, является ли оно простым.

Для решения этой задачи будем перебирать делители числа n . Несложно показать, что для решения задачи достаточно перебирать только делители, не

превосходящие \sqrt{n} , поэтому вычислительная сложность данного алгоритма составляет $O(\sqrt{n})$.