Алгоритмы и их сложность

Алексей Владыкин

СПбГУ ИТМО

28 сентября 2009

Определения алгоритма

- «Алгоритм — это точное предписание, определяющее вычислительный процесс, идущий от варьируемых исходных данных к искомому результату» (А. Марков)
- «Алгоритм — это последовательность действий, либо приводящая к решению задачи, либо поясняющая почему это решение получить нельзя»
- «Алгоритм — это понятные и точные предписания исполнителю совершить конечное число шагов, направленных на решение поставленной задачи»
- Исполнитель алгоритма: машина Тьюринга, компьютер, человек

- Временна́я сложность:
  количество элементарных операций, совершаемых программой

- Пространственная сложность:
  количество ячеек памяти, требуемых программе

O-нотация

- Определение:

$$f(x) = O(g(x)) \quad \Leftrightarrow \quad \forall x \geq x_0 : \ |f(x)| \leq c \cdot |g(x)|$$

- Свойства:
  $f(x) = O(f(x))$
  $c \cdot O(f(x)) = O(f(x))$
  $O(f(x)) \cdot O(g(x)) = O(f(x) \cdot g(x))$

Градации сложности

- $O(1)$ — константная
- $O(\log n)$ — логарифмическая
- $O(n)$ — линейная
- $O(n^a)$ — полиномиальная (квадратичная, кубическая. . .)
- $O(a^n)$ — экспоненциальная
- $O(n!)$ — «факториальная»

Чем меньше — тем лучше?

На практике следует учитывать:

- время реализации и отладки алгоритма;
- константу, скрытую в O-обозначении.

Задача 1

Вычислить сумму

$$1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

Задача 1

Вычислить сумму

$$1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

Первый алгоритм:

- считаем каждый член суммы независимо, затем суммируем;
- сложность $O(n^2)$.

Задача 1

Вычислить сумму

$$1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$$

Первый алгоритм:

- считаем каждый член суммы независимо, затем суммируем;
- сложность $O(n^2)$.

Второй алгоритм:

- k-й член суммы получаем из $(k-1)$-го делением на k;
- сложность $O(n)$.

Задача 2

Вычислить n-е число Фибоначчи:

$$f_1 = 1, \quad f_2 = 1, \quad f_k = f_{k-2} + f_{k-1}$$

Задача 2

Вычислить n-е число Фибоначчи:

$$f_1 = 1, \quad f_2 = 1, \quad f_k = f_{k-2} + f_{k-1}$$

Первый алгоритм:

- считаем рекурсивно «с конца» $(f_n, f_{n-1}, \dots)$;
- сложность $O(2^n)$.

Задача 2

Вычислить n-е число Фибоначчи:

$$f_1 = 1, \quad f_2 = 1, \quad f_k = f_{k-2} + f_{k-1}$$

Первый алгоритм:

- считаем рекурсивно «с конца» $(f_n, f_{n-1}, \dots)$;
- сложность $O(2^n)$.

Второй алгоритм:

- считаем в цикле «с начала» $(f_1, f_2, \dots)$;
- сложность $O(n)$.

Задача 3

Вычислить a^n.

Задача 3

Вычислить a^n.

Первый алгоритм:

- считаем в цикле a, a^2, a^3, $\ldots$;
- сложность $O(n)$.

Задача 3

Вычислить a^n.

Первый алгоритм:

- считаем в цикле a, a^2, a^3, $\ldots$;
- сложность $O(n)$.

Второй алгоритм:

- считаем, используя свойство $a^{2k} = (a^2)^k$;
- сложность $O(\log_2 n)$.

Быстрое возведение в степень

Важные понятия:

- дихотомия — деление пополам;
- инвариант — условие, сохраняющее истинность в процессе работы алгоритма.

Алгоритм:

$$a^n = p \cdot b^q: \quad p = 1, \quad b = a, \quad q = n;$$

$$p \cdot b^q = \begin{cases} (p \cdot b) \cdot (b^2)^{(q-1)/2}, & \text{при нечетных } q, \\ p \cdot (b^2)^{q/2}, & \text{при четных } q. \end{cases}$$

Выход при $q = 0$.

Задача 4

Вычислить значение полинома:

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

Задача 4

Вычислить значение полинома:

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

Первый алгоритм:

- считаем каждый член независимо — сложность $O(n^2)$.

Задача 4

Вычислить значение полинома:

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

Первый алгоритм:

- считаем каждый член независимо — сложность $O(n^2)$.

Второй алгоритм:

- считаем каждый член быстрым возведением в степень — сложность $O(n \log_2 n)$.

Задача 4

Вычислить значение полинома:

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0.$$

Первый алгоритм:

- считаем каждый член независимо — сложность $O(n^2)$.

Второй алгоритм:

- считаем каждый член быстрым возведением в степень — сложность $O(n \log_2 n)$.

Третий алгоритм:

- считаем следующий член на основе предыдущего — сложность $O(n)$.

Задача 5

Найти в массиве чисел элемент, встречающийся нечетное число раз.

Задача 5

Найти в массиве чисел элемент, встречающийся нечетное число раз.

Первый алгоритм:

- брать каждый элемент и считать количество его вхождений — сложность $O(n^2)$.

Задача 5

Найти в массиве чисел элемент, встречающийся нечетное число раз.

Первый алгоритм:

- брать каждый элемент и считать количество его вхождений — сложность $O(n^2)$.

Второй алгоритм:

- за один проход составить таблицу частот, по ней найти ответ — сложность $O(n)$.

Задача 5

Найти в массиве чисел элемент, встречающийся нечетное число раз.

Первый алгоритм:

- брать каждый элемент и считать количество его вхождений — сложность $O(n^2)$.

Второй алгоритм:

- за один проход составить таблицу частот, по ней найти ответ — сложность $O(n)$.

Третий алгоритм:

- Прохорить все элементы массива — сложность $O(n)$.

Задача 6

Проверить наличие числа a в отсортированном массиве чисел.

Задача 6

Проверить наличие числа a в отсортированном массиве чисел.

Первый алгоритм:

- последовательно просмотреть все элементы массива;
- сложность $O(n)$.

Задача 6

Проверить наличие числа a в отсортированном массиве чисел.

Первый алгоритм:

- последовательно просмотреть все элементы массива;
- сложность $O(n)$.

Второй алгоритм:

- двоичный поиск (дихотомия);
- сложность $O(\log_2 n)$.