Асимптотика

Урок 1.1

ПРИ ПОДДЕРЖКЕ
ФОНДА
ПРЕЗИДЕНТСКИХ
ГРАНТОВ

Решение задачи в спортивном программировании: основные моменты_

Решение задачи — программа (исходный код), которая:

- Считывает данные с консоли
- Производит некоторые вычисления
- Печатает ответ на экран

Решение задачи в спортивном программировании: основные моменты_

Необходимое условие — эффективность:

- Программа должна завершать работу в течение заданного количества секунд (time limit)
- Компьютер успевает делать порядка нескольких сотен миллионов операций в секунду

Измерение времени работы_

Напрашивающийся способ — количество базовых действий, выполненных программой

Недостатки:

- Сильная зависимость от входных данных;
- Зависимость от компилятора
- Различная скорость «железа»

Измерение времени работы_

Основные принципы:

- Используем верхнюю оценку на точное количество действий
- Функция эффективности измеряется с точностью до мультипликативной константы: не различаем функции $f(x) = 5*2n$ от $g(x) = 28*2n$
- Главное — порядок роста функции эффективности!

Асимптотическая сложность: формальные определения_

Рассмотрим функции натурального аргумента, т.е. числовые последовательности $f : N \to N$ и $g : N \to N$;

1. $f = O(g)$ (“O большое от g”), если $\exists\, C_0, n_0 \forall\, n \geq n_0 : f(n) \leq C_0\, g(n)$;

Асимптотическая сложность: формальные определения_

2. $f = \Omega(g)$ (“Омега большое от g”), если $\exists\ C_0 > 0,\ n_0 : \forall\ n \geq n_0$: $f(n) \geq C_0\ g(n)$;

3. $f = \Theta(g)$(“тета большое от g”), если $f = O(g)$ и $f = \Omega(g)$, т.е. $\exists\ C_1 > 0\ ,\ C_2,\ n_0 \forall\ n \geq n_0 : C_1 g(n) \leq f(n) \leq C_2 g(n)$; в этом случае функции f и g «асимптотически эквивалентны»

Асимптотика: примеры_

1. $f(n) = n$, $g(n) = n^2$. Тогда $f = O(g)$, но $g \,!= O(f)$;

2. $f(n) = 5*n$, $g(n) = 7*n$. Тогда $f = \Theta(g)$.

3. $f(n) = 1000000000n$. Тогда $f(n) = O(n^2)$.

4. $f(n) = 8*n^2 + 36 * n + 4360$. Тогда $f = \Theta(n^2)$, $f = O(n^2)$.

Асимптотика: примеры_

5. $f(n) = \log_a n$, $g(n) = \log_b n$, $a, b > 1$. Тогда $f = \Theta(g)$,
   ибо $g(n) / f(n) = \log_b a = const$

6. $f(n) = \ln^k n$. Тогда $\forall l > 0$: $f = O(n^l)$. При этом, $f \mathrel{!=} O(n^0)$

7. $f(n) = n^k$, $k > 0$. Тогда $f(n) = O(2^n)$, $O(2^n n)$, $O(3^n)$ ...

Измерение времени работы_

Основные принципы:

- Используем верхнюю оценку на точное количество действий
- Функция эффективности измеряется с точностью до мультипликативной константы: не различаем функции $f(x) = 5*2^n$ от $g(x) = 28*2^n$
- Главное — порядок роста функции эффективности!

Практическая оценка времени работы_

На практике, эффективность оценивается следующим образом:

- Вычисляется асимптотика алгоритма, например $O(f(n))$
- Игнорируем константу; в f подставляются максимально возможные значения; получаем некоторое число C

Практическая оценка времени работы_

Психологическая граница — 10^8:

- Если C значительно меньше — программа сработает быстро
- Если C значительно больше — программа почти наверное не уложится в секунду-две
- Если C в районе 10^8 — как повезет

Окончание_

Мы узнали:

- Что такое задача в спортивном программировании и ее решение
- Что такое асимптотика, сложность алгоритма
- Как оценивать, достаточно ли быстр алгоритм

В следующих уроках мы рассмотрим примеры базовых алгоритмов и их сложностей

ПРИ ПОДДЕРЖКЕ ФОНДА ПРЕЗИДЕНТСКИХ ГРАНТОВ