Алгоритмы и структуры данных

Асимптотический анализ. Рекуррентные соотношения



Практическое занятие **03**

2026–2027 учебный год

План

1. Проверка **асимптотических** границ: применение определений символов Ландау
2. Рекуррентные соотношения: анализ временной сложности **рекурсивных** алгоритмов
3. Практический анализ функций временной сложности

Warm-Up

FAQ Асимптотический анализ

1. Если $T(n) = \Theta\big(f(n)\big)$, то что еще верно?
2. $\mathrm{O}\big(f(n)\big)$ определяет множество функций.
   Какова его мощность?
3. Какая функция растет быстрее – $\sqrt{n}$ или $\log n$?
4. Рекуррентное соотношение определяет...
5. Что означает найти решение рекуррентного соотношения?

Манипуляции с асимптотикой

Важно





Использование O, Θ, Ω в выражениях

1. «Злоупотребление» знаком равенства [abuse of notation]
   $T(n) \in \Theta(g(n))$ считается равносильной записи $T(n) = \Theta(g(n))$

2. Выполнение арифметических операций с асипмтотическими нотациями потребует перехода к функциям:
   $O(n) + \Theta(n^2) + \Omega(1) = ???$

Важно





Использование O, Θ, Ω в выражениях

1. «Злоупотребление» знаком равенства [abuse of notation]
   $T(n) \in \Theta(g(n))$ считается равносильной записи $T(n) = \Theta(g(n))$

2. Выполнение арифметических операций с асипмтотическими нотациями потребует перехода к функциям:
   $O(n) + \Theta(n^2) + \Omega(1) = T_1(n) + T_2(n) + T_3(n)$, где
   - $0 \leq T_1(n) \leq an$
   - $bn^2 \leq T_2(n) \leq cn^2$
   - $0 \leq d \leq T_3(n)$

   для некоторых констант $a, b, c, d > 0$.

Важно





Использование O, Θ, Ω в выражениях

1. «Злоупотребление» знаком равенства [abuse of notation]
   $T(n) \in \Theta(g(n))$ считается равносильной записи $T(n) = \Theta(g(n))$

2. Выполнение арифметических операций с асипмтотическими нотациями потребует перехода к функциям:
   $O(n) + \Theta(n^2) + \Omega(1) = T_1(n) + T_2(n) + T_3(n)$, где

   $$\left.\begin{array}{l} 0 \le T_1(n) \le an \\ bn^2 \le T_2(n) \le cn^2 \\ 0 \le d \le T_3(n) < \infty \end{array}\right\} \Rightarrow T_1(n) + T_2(n) + T_3(n) \ge bn^2 + d$$

   для некоторых констант $a, b, c, d > 0$.

Важно





Использование O, Θ, Ω в выражениях

1. «Злоупотребление» знаком равенства [abuse of notation]
   $T(n) \in \Theta(g(n))$ считается равносильной записи $T(n) = \Theta(g(n))$

2. Выполнение арифметических операций с асипмтотическими нотациями потребует перехода к функциям:
   $O(n) + \Theta(n^2) + \Omega(1) = T_1(n) + T_2(n) + T_3(n)$, где

   - $0 \le T_1(n) \le an$
   - $bn^2 \le T_2(n) \le cn^2$
   - $0 \le d \le T_3(n) < \infty$

   $\Rightarrow T_1(n) + T_2(n) + T_3(n) = \Omega(n^2)$

   строго говоря, про верхнюю границу суммы этих трех функций сказать ничего определенного нельзя

   для некоторых констант $a, b, c, d > 0$.

Важно




Использование O, Θ, Ω в выражениях

1. «Злоупотребление» знаком равенства [abuse of notation]
   $T(n) \in \Theta(g(n))$ считается равносильной записи $T(n) = \Theta(g(n))$

2. Выполнение арифметических операций с асипмтотическими нотациями потребует перехода к функциям:
   $O(n) + \Theta(n^2) + \Omega(1) = T_1(n) + T_2(n) + T_3(n)$, где

   - $0 \le T_1(n) \le an$
   - $bn^2 \le T_2(n) \le cn^2$
   - $0 \le d \le T_3(n) \le en^2$

   $\Rightarrow T_1(n) + T_2(n) + T_3(n) = \Theta(n^2)$

   на практике, часто также подразумевается, что $T_3(n) \in O(n^2)$ — главный по порядку роста терм «не перебивается»

   для некоторых констант $a, b, c, d, e > 0$.

Задача 1





Свойства асимптотических нотаций

Всегда ли верны следующие утверждения?

1. Сумма: $O\big(f(n) + g(n)\big) = O\big(f(n)\big) + O\big(g(n)\big)$.
2. Произведение: $O\big(f(n) \cdot g(n)\big) = O\big(f(n)\big) \cdot O\big(g(n)\big)$.
3. Если $g(n) = O\big(f(n)\big)$ и $h(n) = O\big(f(n)\big)$, то $g(n) = O\big(h(n)\big)$.
4. Если $f(n) = O\big(g(n)\big)$, то $\log_2 f(n) = O(\log_2 g(n))$ при $\log_2 g(n) \geq 1$ и $f(n) \geq 1$.

Задача 2



Применение асимптотических нотаций

1. Покажите, что для любых a и b, где $b > 0$, верно, что $(n + a)^b = O(n^b)$.
2. Оцените осмысленность следующего утверждения: «*Время выполнения некоторого алгоритма как минимум* $O(n^2)$».
3. Верно ли, что $2^{n+1} = O(2^n)$ и $2^{2n} = O(2^n)$?

Задача 3





Взаимная асимптотика

Сопоставьте следующие функции временной сложности по их асимптотическому поведению.

$T_1(n)$	$T_2(n)$	$T_1(n) = O(T_2(n))$?	$T_1(n) = \Omega(T_2(n))$?	$T_1(n) = \Theta(T_2(n))$?
$25n \ln n + 5n$	$0.5n \log_2 n$			
$\sqrt{n} \log_2 n$	n			
$n \sqrt{n}$	$n^{1.4}$			

Задача 4





Известное рекуррентное соотношение

Обоснуйте асимптотическую верхнюю границу временной сложности алгоритма, работа которого характеризуется следующим рекуррентным соотношением:

$$T(n) = \begin{cases} 2 \cdot T(n-1) - 1 & \text{при } n > 0 \\ 1 & \text{иначе.} \end{cases}$$

Решите рекуррентное соотношение методом итераций и методом мат. индукции [подстановки].

Задача 5

Неизвестное рекуррентное соотношение

RECURSIVE ALGORITHM.cpp

```cpp
int func(int n) {
    if (n <= 2) return 1;

    int  b = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 1; j < n; j <<= 1) {
            ++b;
        }
    }

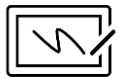
    return func(n - 2) + b;
}
```

1. Составить рекуррентное соотношение для $T(n)$ — временной сложности функции func.
2. Найти возможную асимптотическую верхнюю границу для $T(n)$.

Практический анализ асимптотики

Задача 6





Сравнение сложности алгоритмов

1. Время работы алгоритма **A** (в микросекундах) зависит от размера входных данных следующим образом:
   $\boldsymbol{T_A(n) = 0,1n^2 \log_{10} n}$.
2. Время работы алгоритма **B** (в микросекундах) зависит от размера входных данных следующим образом:
   $\boldsymbol{T_B(n) = 2,5n^2}$.

- Какой из этих алгоритмов более эффективен с точки зрения O? Начиная с какого размера входных данных?
- Если практический размер входных данных не превышает 10^9, какому алгоритму следует отдать предпочтение?

Задача 7



Сравнение сложности алгоритмов

1. Время работы алгоритма **A** (в миллисекундах) зависит от размера входных данных следующим образом:
   $\boldsymbol{T_A(n) = c_A n \log_{10} n}$.
2. Время работы алгоритма **B** (в микросекундах) зависит от размера входных данных следующим образом:
   $\boldsymbol{T_B(n) = c_B n}$.

- Практические тесты показали, что обработка $\boldsymbol{10^4}$ объектов алгоритмом **A** занимает **100 миллисекунд**, а алгоритмом **B** – **500 миллисекунд**.
- При каких размерах входных данных алгоритм **B** более эффективен?
- Если практический размер входных данных не превышает 10^9, какому алгоритму следует отдать предпочтение?

Задача 8

Верхняя граница временной сложности

function.cpp

```cpp
void func(int n) {
    int i = 1, s = 1;
    while (s <= n) {
        ++i;
        s += i;
    }
}
```

Оцените и обоснуйте асимптотическую верхнюю границу сложности для этой функции в соответствии с определением O.

Резюме

1. Взаимосвязь **асимптотических** границ точной функции временной сложности алгоритмов

2. Базовые подходы к решению **рекуррентных** соотношений

3. Сопоставление **теоретических** и **практических** оценок временной сложности алгоритмов