CS 240 Online Fall 2020

Lin Xuntian

September 2020

Contents

1 Introduction and Asymptotic Analysis

1.1 Asymptotic Notation

Definition

O-notation: $f(n) \in O(g(n))$ if there exists constants $c > 0$ and $n_0 > 0$ such that $|f(n)| \leqslant c|g(n)|$ for all $n \geqslant n_0$.

Example 1

Prove $(n+1)^5$ is $O(n^5)$.

Solution: Clearly, $g(n)$ will never dominates $f(n)$.

Observe,

$$\begin{aligned} n+1 &\leqslant 2n \quad \text{for } n \leqslant 1 \\ (n+1)^5 &\leqslant (2n)^5 = 2^5 n^5 = 32n^5 \quad \forall n \leqslant 1 \end{aligned}$$

From First Principles means explicitly find c and n_0. Sometimes, we just want to prove they exist.

Example 2

Properties: Assume $f(n) \leqslant 0$, $g(n) \leqslant 0$ for all $n \leqslant 0$.

1. $af(n)$ is $O(f(n))$ for all constants $a > 0$
2. If $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$ then $f(n)$ is $O(h(n)$.
3. $\max(f(n), g(n))$ is $O(f(n) + g(n))$
4. $a_0 + a_1 n + \cdots + a_k n^k$ is $O(n^k)$, $a_i > 0$
5. $n^x \in O(a^n),\ x > 0,\ a > 1$
6. $(\log n)^x \in O(n^y)$ $x, y > 0$

Proof:

1. $c = a,\ n_0 = 1$
2. $$\begin{aligned} f(n) &\leqslant c_1 g(n) \quad \forall n \geqslant n_1 \\ g(n) &\leqslant c_2 h(n) \quad \forall n \geqslant n_2 \\ f(n) &\leqslant c_1 c_2 h(n) \quad \forall n \geqslant \max(n_1, n_2) \end{aligned}$$

   Then, $f(n) \leqslant ch(n) \quad \forall n \geqslant n_0$ where $c = c_1 c_2$ and $n_0 = \max(n_1, n_2)$

Definition

Ω-notation: $f(n) \in \Omega(g(n))$ if there exists constants $c > 0$ and $n_0 > 0$ such that $c|g(n)| \leqslant |f(n)|$ for all $n \geqslant n_0$.

Definition

Θ-notation: $f(n) \in \Theta(g(n))$ if there exists constants $c_1, c_2 > 0$ and $n_0 > 0$ such that $c|g(n)| \leqslant |f(n)|$ for all $n \geqslant n_0$.

Example 3 Show $n^2 + n\log n + n \in O(n^2)$.

Since $\log n \leqslant n \ \forall n \geqslant 1$, then

$$\begin{aligned} n^2 + \log n + n &\leqslant n^2 + n^2 + n^2 \quad \forall n \geqslant 1 \\ &= 3n^2 \quad \forall n \geqslant 1 \end{aligned}$$

Example 4 Show $n^3 \log n \in \Omega(n^3)$.

$$1n^3 \leqslant n^3 \log n \quad \forall n \geqslant 1$$

Definition o-notation: $f(n) \in o(g(n))$ if for all constants $c > 0$, there exists a constant $n_0 > 0$ such that $|f(n)| \leqslant c|g(n)|$ for all $n \geqslant n_0$.

Example 5 Show $2020n^2 + 1388n \in o(n^3)$.

Proof: Let $c > 0$ be given, $0 < c < 1$ is possible.

$$\begin{aligned} 2020n^2 + 1388n &< 5000n^2 \quad \forall n \geqslant 1 \\ &= \frac{5000}{n} n^3 \\ &\leqslant cn^3 \end{aligned}$$

Note: $\dfrac{5000}{n} \leqslant c$, $\forall n \geqslant \dfrac{5000}{c}$

n_0 may depend on given c, can go out as far as required. So for any c, choose $n_0 = \dfrac{5000}{c}$.

1.2 Analysis of Algorithms

1.2.1 Complexity of Algorithms

Definition The **worst-case complexity of an algorithm** $\mathcal{A}$ is a function $f\colon \mathbb{Z}^+ \to \mathbb{R}$ mapping n (the input size) to the longest running time for any input instance of size n:

$$T_{\mathcal{A}(n)} = \max\{T_{\mathcal{A}}(I)\colon \text{Size}(I) = n\}.$$

2 Priority Queues

2.1 Binary Heaps

Definition

A **heap** A heap is a binary tree with the following two properties:

1. **Structural Property:** All the levels of a heap are completely filled, except (possibly) for the last level. The filled items in the last level are left-justified.
2. **Heap-order Property:** For any node i, the key of the parent of i is larger than or equal to key of i.

The full name for this is max-oriented binary heap.