BACS2063 Data Structures and Algorithms

Efficiency of Algorithms Extra Reading

Chapter 2

Formalities

- Formal definition of Big O
  An algorithm's time requirement **f(n)** is of order at most **g(n)**
  - **f(n) = O(g(n))**
  - For a positive real number ***c*** and positive integer ***N*** exist such that

$$f(n) \leq c \cdot g(n) \quad \text{for all } n \geq N$$

Formalities

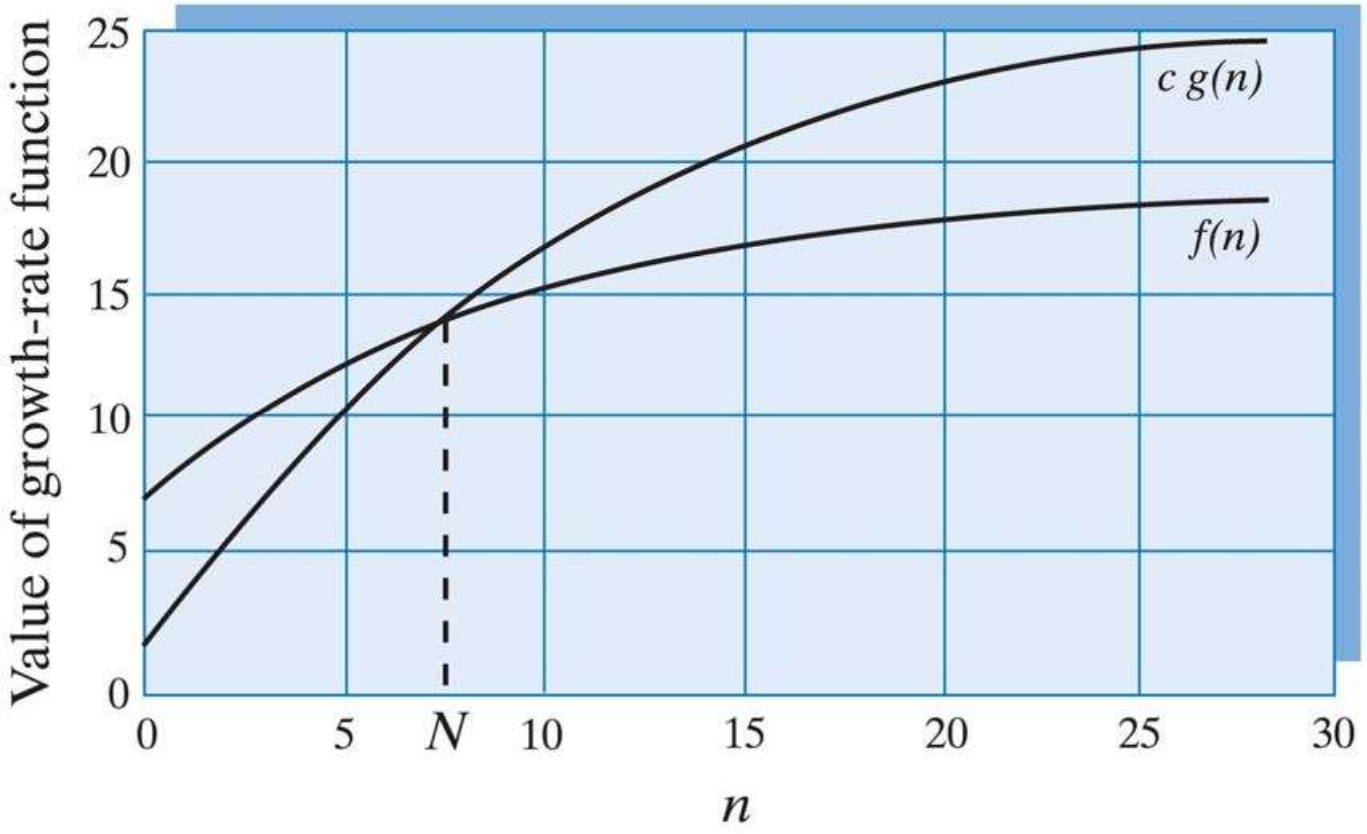

Fig. 2.7: An illustration of the definition of Big O

Formalities

- The following identities hold for Big O notation:
  1. **O(k f(n)) = O(f(n)) for a constant k**
  2. **O(f(n)) + O(g(n)) = O(f(n) + g(n))**
  3. **O(f(n)) O(g(n)) = O(f(n) g(n))**