CSE 332: Data Structures

Asymptotic Analysis

Richard Anderson, Steve Seitz

Winter 2014

Key ideas

- Express runtime as a function of input size
  - E.g., T(n) is the maximum runtime of the algorithm for inputs of size n
- Constant factors don’t matter in runtime analysis
  - Constants depend on machine model
  - Very tedious to determine
- Import case is for large n
  - We study how runtime increases for large inputs

Definition of Order Notation

- $h(n) \in O(f(n))$ Big-O "Order"

  if there exist positive constants c and n_0

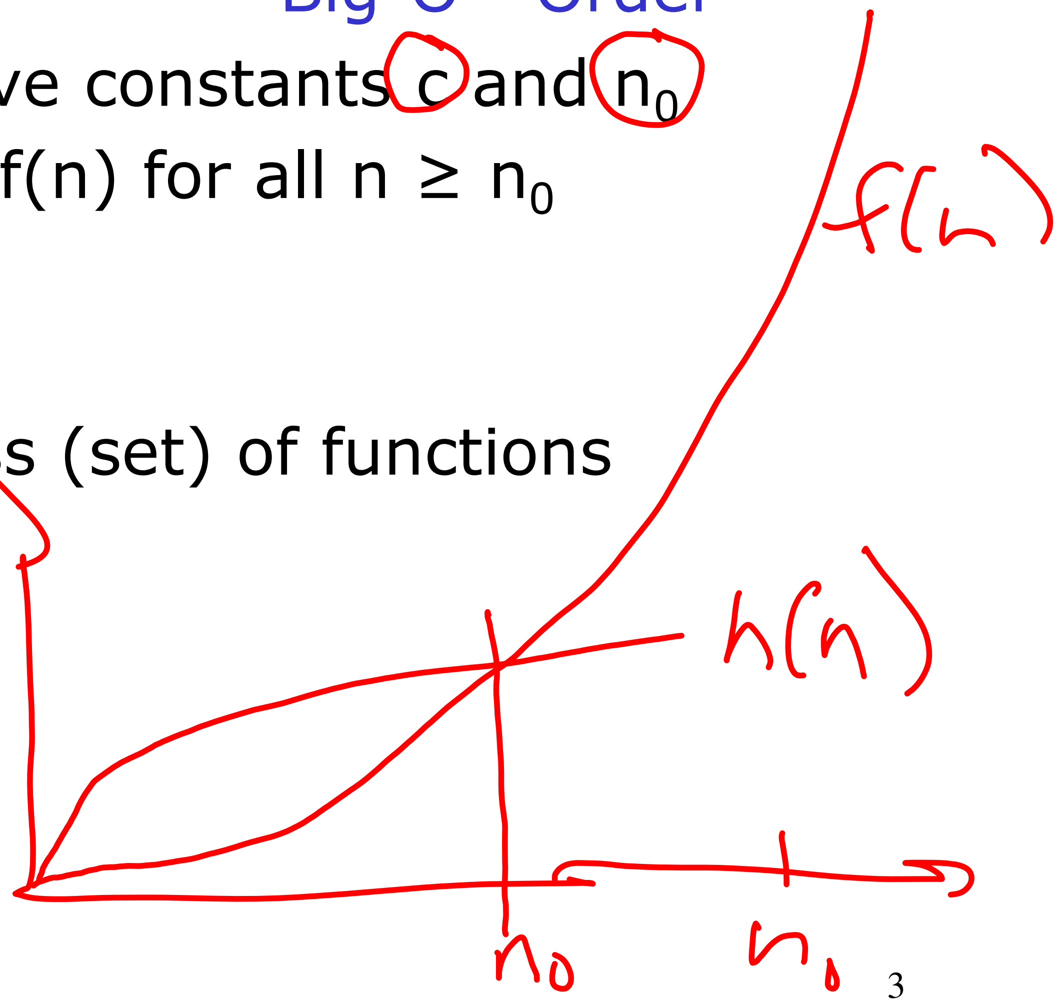
  such that $h(n) \leq c\, f(n)$ for all $n \geq n_0$

O(f(n)) defines a class (set) of functions

$h(n) = 5 \log n \in O(\log n)$

$c = 10$

$5 \log n < 10 \log n$



$\sqrt{n} \in \Omega(\log n)$

Asymptotic Lower Bounds

- $\Omega(\ g(n)\ )$ is the set of all functions asymptotically greater than or equal to $g(n)$

$n^2 \in \Omega(n \log n)$

- $h(n) \in \Omega(\ g(n)\ )$ iff
  There exist $c>0$ and $n_0>0$ such that $h(n) \geq c\, g(n)$ for all $n \geq n_0$

Asymptotic Tight Bound

- $\theta(\ f(n)\ )$ is the set of all functions asymptotically equal to $f\ (n)$

- $h(n) \in \theta(\ f(n)\ )$ iff
  $h(n) \in O(\ f(n)\ )$ and $h(n) \in \Omega(f(n)\ )$
  - This is equivalent to:

$$\lim_{n\to\infty} h(n)/f(n) = c \neq 0$$

$h(n) = 5\log n + 3 \in \Theta(\log n)$

Example

- $F(n) = 4n^2 + n \log n$

- $F(n)$ is $O(n^3)$? Yes
- $F(n)$ is $\Omega(n^3)$? No
- $F(n)$ is $\Omega(n^2)$? Yes
- $F(n)$ is $\theta(n^2)$? Yes
- $F(n)$ is $\theta(n^3)$? No

Full Set of Asymptotic Bounds

- O($f(n)$) is the set of all functions asymptotically less than or equal to $f(n)$
  - o($f(n)$) is the set of all functions asymptotically strictly less than $f(n)$

- Ω($g(n)$) is the set of all functions asymptotically greater than or equal to $g(n)$
  - ω($g(n)$) is the set of all functions asymptotically strictly greater than $g(n)$

- θ($f(n)$) is the set of all functions asymptotically equal to $f(n)$

Formal Definitions

- $h(n) \in O(\ f(n)\ )$ iff
  There exist $c>0$ and $n_0>0$ such that $h(n) \leq\ c\, f(n)$ for all $n \geq n_0$

- $h(n) \in o(f(n))$ iff
  There exists an $n_0>0$ such that $h(n)$ **<** $c\, f(n)$ for all $c>0$ and $n \geq n_0$
  - This is equivalent to: $\lim_{n\to\infty} h(n)/f(n) = 0$

- $h(n) \in \Omega(\ g(n)\ )$ iff
  There exist $c>0$ and $n_0>0$ such that $h(n) \geq c\, g(n)$ for all $n \geq n_0$

- $h(n) \in \omega(\ g(n)\ )$ iff
  There exists an $n_0>0$ such that $h(n)$ **>** $c\, g(n)$ for all $c>0$ and $n \geq n_0$
  - This is equivalent to: $\lim_{n\to\infty} h(n)/g(n) = \infty$

- $h(n) \in \theta(\ f(n)\ )$ iff
  $h(n) \in O(\ f(n)\ )$ and $h(n) \in \Omega(f(n)\ )$
  - This is equivalent to: $\lim_{n\to\infty} h(n)/f(n) = c \neq 0$

Big-Omega et al. Intuitively

Asymptotic Notation	Mathematics Relation
O	$\leq$
Ω	$\geq$
θ	$=$
o	$<$
ω	$>$

Complexity cases (revisited)

Problem size **N**

- **Worst-case complexity**: **max** # steps algorithm takes on "most challenging" input of size **N**
- **Best-case complexity: min** # steps algorithm takes on "easiest" input of size **N**

- **Average-case complexity**: **avg** # steps algorithm takes on *random* inputs of size **N**
- **Amortized complexity**: **max** total # steps algorithm takes on **M** "most challenging" *consecutive* inputs of size **N**, divided by **M** (i.e., divide the max total by **M**).