AE2ADS: Algorithms Data Structures and Efficiency

Lecturer: Heshan Du
Email: Heshan.Du@nottingham.edu.cn

University of Nottingham Ningbo China

Big-Oh

Let $f(n)$ and $g(n)$ be functions mapping positive integers to positive real numbers.

We say that $f(n)$ is $O(g(n))$, if there exist a real constant $c > 0$ and an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$,

$$f(n) \leq cg(n).$$

Exercise 1

Prove that:

- $n^2 + 1$ is $O(n^2)$
  - $c = 2, n_0 = 1$
- $(n - 3)^2$ is $O(n^2)$
  - $c = 10, n_0 = 1$

Exercise 2

Given that $f(n) = n + 3$, if n is even; $f(n) = n^2 + 5$, if n is odd, state the Big-Oh behaviour of $f(n)$, and prove it.

Hint: $f(n)$ is $O(n^2)$. Let $c = 6, n_0 = 1$.

Big-Omega

Let $f(n)$ and $g(n)$ be functions mapping positive integers to positive real numbers.

We say that $f(n)$ is $\Omega(g(n))$, if there exist a real constant $c > 0$ and an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$,

$$f(n) \geq cg(n).$$

Big-Theta

Let $f(n)$ and $g(n)$ be functions mapping positive integers to positive real numbers.

We say that $f(n)$ is $\Theta(g(n))$, if there are real constants $c' > 0$, $c'' > 0$, and an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$,

$$c'g(n) \leq f(n) \leq c''g(n).$$

Little-Oh

Let $f(n)$ and $g(n)$ be functions mapping positive integers to positive real numbers.

We say that $f(n)$ is $o(g(n))$, if for every real constant $c > 0$, there exists an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$,

$$f(n) < cg(n).$$

Exercise 3

- Prove or disprove that:
  1. 5 is $\Omega(1)$
  2. $2n + 1$ is $\Omega(n)$
  3. 5 is $\mathrm{o}(1)$
  4. 5 is $\mathrm{o}(n)$
  5. $n^2 - 5n$ is $\Theta(n^2)$
  6. n^2 is $\Omega(n)$
  7. 1 is $\mathrm{o}(\log n)$
  8. $n \log n$ is $\mathrm{o}(n^2)$

Exercise 3

Prove or disprove that:

1. 5 is $\Omega(1)$ [hint: yes, $c = 1, n_0 = 1$]
2. $2n + 1$ is $\Omega(n)$ [hint: yes, $c = 1, n_0 = 1$]
3. 5 is $\mathrm{o}(1)$ [hint: no, e.g., when $c = 1$]
4. 5 is $\mathrm{o}(n)$ [hint: yes, n_0 is an integer greater than $\frac{5}{c}$]
5. $n^2 - 5n$ is $\Theta(n^2)$ [hint: yes, $c'' = 1, c' = 0.5, n_0 = 10$]
6. n^2 is $\Omega(n)$ [hint: yes, $c = 1, n_0 = 1$]
7. 1 is $\mathrm{o}(\log n)$ [hint: yes, n_0 is an integer greater than $2^{\frac{1}{c}}$]
8. $n \log n$ is $\mathrm{o}(n^2)$ [hint: yes]

Exercise 3.7

Prove 1 is o$(\log n)$.

Proof: In order to show 1 is $o(\log n)$, by the definition of little o, take an arbitrary positive real constant c, we need to show that there exists an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$, 1$< c(\log n)$, this is, $\frac{1}{c} < \log n$, $2^{\frac{1}{c}} < n$. Let n_0 be the smallest integer that is ≥ 1 and >$2^{\frac{1}{c}}$. Then it is clear that, for every $n \geq n_0 > 2^{\frac{1}{c}}$, $1 < c(\log n)$ is true. Hence, 1 is o$(\log n)$.

Exercise 3.8

Prove $n \log n$ is o(n^2)

Proof: In order to show $n \log n$ is o(n^2), by the definition of little o, take an arbitrary positive real constant c, we need to show that there exists an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$, $n \log n < \ c(n^2)$, this is, $c > (\log n)/n$. As $n \ \rightarrow +\infty$, $(\log n)/n \rightarrow 0$. This means that it eventually will be smaller than c. Hence there exists an integer constant $n_0 \geq 1$ such that for every $n \geq n_0$, $n \log n < \ c(n^2)$. Hence, 1 is o$(\log n)$.

Exercise 4

Given $f(n) = n^2$ if n is even, $f(n) = n$ if n is odd. Find the big-Oh and big-Omega behaviors of $f(n)$.

Hint: $f(n) \in O(n^2)$; $f(n) \in \Omega(n)$

More Exercises

M. T. Goodrich, R. Tamassia and M. H. Goldwasser, *Data Structures and Algorithms in Java*, 6th Edition, 2014.

– Chapter 4. Analysis Tools