COP 3503
Lecture Notes

Mark Williams

Contents

Preface

Chapter 1

Exam 1

1.1 Big-Oh

1.1.1 The formal definition of Big-Oh

Big-Oh is the way in which Computer Scientists mathematically question the running times of our programs. It allows a general understanding how an algorithm will work with large, to infinitely large datasets or inputs.

Definition 1.1: Big-Oh, Big-Ω and Big-Θ

Let c, N represent constants where c is some constant towards the functions, $c_1 > 0$, and N be where $cg(n)$ meets or exceeds $f(n)$.

Big-Oh:

$$f(n) = O(g(n)) \text{ iff } f(n) \leq c_1 g(n) \text{ for } n \geq N_0$$

Big-Ω:

$$f(n) = \Omega(g(n)) \text{ iff } f(n) \geq c_1 g(n) \text{ for } n \geq N_0$$

Big-Θ:

$$f(n) = \Theta(g(n)) \text{ iff: } \begin{cases} f(n) = O(g(n)) \\ \wedge \\ f(n) = \Omega(g(n)) \end{cases}$$

Big-Oh is used to find an **upper bound** of a function. Big-Ω is used to find a **lower bound** of a function.