CS3230 Semester 1 2024/2025

Design and Analysis of Algorithms

Tutorial 01
Introduction and Asymptotic Analysis
For Week 02

Document is last modified on: July 24, 2024

1 Notes

CS3230 tutorial format is as follows: We will consider a few questions per tutorial. Some questions are **revealed beforehand** (on Canvas), some are **hidden** and will only be discussed on the spot.

For **each question**, we will ask a student to solve it. A **reasonable** attempt for that question will earn the student one participation point (1%). The **limit is maximum 3 points (3%)** for a student for the whole semester. TA will try to ensure that each student do at least one question throughout the semester.

Note that since this is the first tutorial, your TA will start the session with a short icebreaker.

2 Lecture Review: Asymptotic Analysis

We say $f \in O(g)$ or $f = O(g)$ or $f(n) = O(g(n))$ if $\exists c, n_0 > 0$ such that $\forall n \geq n_0, 0 \leq f(n) \leq c \times g(n)$. Informally, we say (function) g is an upperbound on (function) f. This is the most popular Big O worst-case time complexity analysis that we have learned since earlier course, i.e., from CS2040/C/S.

Copy-pasting similar mathematical statement four other times for the other asymptotic notations $\Omega, \Theta, o, \omega$ is probably less clear compared to the following tabular summary:

We say	if $\exists c, c_1, c_2, n_0 > 0$ such that $\forall n \geq n_0$	In other words
$f(n) = O(g(n))$	$0 \leq f(n) \leq c \times g(n)$	g is an **upper** bound on f
$f(n) = \Omega(g(n))$	$0 \leq c \times g(n) \leq f(n)$	g is a **lower** bound on f
$f(n) = \Theta(g(n))$	$0 \leq c_1 \times g(n) \leq f(n) \times c_2 \times f(n)$	g is a **tight** bound on f
$f(n) = o(g(n))$	$0 \leq f(n) < c \times g(n)$	g is a **strict upper** bound on f
$f(n) = \omega(g(n))$	$0 \leq c \times g(n) < f(n)$	g is a **strict lower** bound on f

3 Tutorial 01 Questions

Q1). Assume $f(n), g(n) > 0$, show:

- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f(n) = o(g(n))$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty \Rightarrow f(n) = O(g(n))$
- $0 < \lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty \Rightarrow f(n) = \Theta(g(n))$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)} > 0 \Rightarrow f(n) = \Omega(g(n))$
- $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \Rightarrow f(n) = \omega(g(n))$

Q2). Assume $f(n), g(n) > 0$, show:

- Reflexivity
  - $f(n) = O(f(n))$
  - $f(n) = \Omega(f(n))$
  - $f(n) = \Theta(f(n))$
- Transitivity
  - $f(n) = O(g(n))$ and $g(n) = O(h(n)$ implies $f(n) = O(h(n))$
  - Do the same for Ω, Θ, o, ω
- Symmetry
  - $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$
- Complementarity
  - $f(n) = O(g(n))$ iff $g(n) = \Omega(f(n))$
  - $f(n) = o(g(n))$ iff $g(n) = \omega(f(n))$

Q3). Which of the following statement(s) is/are True?

1. $3^{n+1} = O(3^n)$
2. $4^n = O(2^n)$
3. $2^{\lfloor \log n \rfloor} = \Theta(n)$ (we assume log is in base 2)
4. For a constant $i, a > 0$, we have $(n + a)^i = O(n^i)$

Q4). Which of the following statement(s) is/are True?
$2^{\log_2 n} =$

1. $O(n)$
2. $\Omega(n)$
3. $\Theta(\sqrt{n})$
4. $\omega(n)$

Q5). Rank the following functions by their order of growth.
(But if any two (or more) functions have the same order of growth, group them together).

- $f_1(n) = \log n$
- $f_2(n) = n!$
- $f_3(n) = 2^n + n$
- $f_4(n) = n^{2.3} + 16n$