CSCI-SHU 220 - Algorithms

Recitation 1

Hand-written notes during recitation will be available at:

https://www.dropbox.com/sh/x1z104c22d51pox/AACiJdDSKe2SDZw3qNljNApka?dl=0

A. Asymptotics: a review

- *Why asymptotics*?) In short:

$$\text{to suppress} \quad \underbrace{\textit{constant factors}}_{\text{machine dependent}} \quad \text{and} \quad \underbrace{\textit{lower-order terms}}_{\text{irrelevant for large inputs}}$$

  Historical point of view: check the paper of Donald E. Knuth: [1]

  > "On the basis of the issues discussed here, I propose that members of SIGACT,[2] and editors of computer science and mathematics journals, adopt the O,Ω, and Θ notations as defined above, unless a better alternative can be found reasonably soon".

- *Big-O notation in English words*:

  $T(n) = O(f(n))$ if and only if $T(n)$ is *eventually bounded above* by a *constant* multiple of $f(n)$.

  Note: by a *constant* multiple we mean some number independent of n.

  A set-theoretical view: $O(f(n))$ is the set of all functions which are eventually bounded above by $f(n)$. By $T(n) = O(f(n))$ it means that $T(n) \in O(f(n))$.[3]

- *Big-O notation in mathemathcal words*:

  $T(n) = O(f(n))$ if and only if there are positive constants c and n_0 such that

$$T(n) \leq c \cdot f(n), \quad \text{for all } n \geq n_0. \tag{1}$$

  A game-theoretical view: if you want to prove $O(f(n))$, then your task is to choose the constants c and n_0 such that (1) holds whenever $n \geq n_0$. Think of it as a contest between you and an opponent. You go first, and have to commit to constants c and n_0. Your opponent goes second and can choose any integer n that is at least n_0. You win if (1) holds; your opponent wins if the opposite inequality $T(n) > c \cdot f(n)$ holds.

B. Exercises

1. Describe Θ and Ω notations in English words.
2. Suppose $T(n) = a_k n^k + \cdots + a_1 n + a_0$, where k is a non-negative integer and the a_i's are real numbers. Prove that $T(n) = O(n^k)$.
3. Let $k \geq 1$ be a positive integer and let $T(n) = n^k$. Prove that $T(n)$ is not $O(n^{k-1})$.
4. For $T(n) = 2^{n+100}$, prove or disprove: $T(n) = O(2^n)$.
5. For $T(n) = 2^{100n}$, prove or disprove: $T(n) = O(2^n)$.
6. Let f and g denote functions from the positive integers to the non-negative real numbers, and define $T(n) = \max(f(n), g(n))$ for each $n > 1$. Prove that $T(n) = \Theta(f(n) + g(n))$.
7. Let f and g be non-decreasing real-valued functions defined on the positive integers, and $f(1) \geq 1$ and $g(1) \geq 1$. Prove or disprove that $f(n) = O(g(n)) \Rightarrow 2^{f(n)} = O(2^{g(n)})$.

[1] Donald E. Knuth, "Big Omicron and Big Omega and Big Theta," SIGACT News, Apr.-June 1976, page 23. Reprinted in *Selected Papers on Analysis of Algorithms* (Center for the study of Language and Information, 2000).

[2] SIGACT is the special interest group of the ACM (Association for Computing Machinery) that concerns theoretical computer science, and in particular the analysis of algorithms.

[3] Before the adoptation of the big-O notation, Knuth used $(\leq f(n))$ to denote the set of all functions which are eventually bounded above by $f(n)$, and you can guess what $(\sim f(n))$ and $(\geq f(n))$ are.

8. Arrange the following functions in order of increasing growth rate, with $g(n)$ following $f(n)$ in your list if and only if $f(n) = O(g(n))$.
   a) $n^2 \log_2 n$
   b) 2^n
   c) 2^{2^n}
   d) $n^{\log_2 n}$
   e) n^2
9. Prove that
$$\log(n!) = \Theta(n \log n). \tag{2}$$
   Hint: To show an upper bound, compare $n!$ with n^n. To show a lower bound, compare it with $(n/2)^{n/2}$.
10. Prove that
$$\sum_{k=1}^{2^n} \frac{1}{k} = \Theta(n). \tag{3}$$

C. The watermelon problem.

Let us go back to the watermelon problem in mathematical terms.

(a) What is the maximum number L_n of 2D regions defined by n lines of $\mathbb{R}^2$? *Examples*: $L_1 = 2, L_2 = 4, L_3 = 7$. *Hint*: find a relation between L_n and L_{n-1}.

(b) What is the maximum number P_n of 3D regions defined by n parallel planes of $\mathbb{R}^3$?

(c) Next we will ask you for the maximum number P_n of 3D regions defined by n planes of $\mathbb{R}^3$. Think the moment when you have $n-1$ planes and you are going to add a new one to obtain the regions defined by those n planes. At most how many lines does the newly added plane contain as the intersections with other $n-1$ planes? What is the maximum number of 2D regions defined by those lines?

(d) What is the maximum number P_n of 3D regions defined by n planes of $\mathbb{R}^3$?

D. LaTeX.

To start with, read this:

https://tobi.oetiker.ch/lshort/lshort.pdf

The following link is useful when you try to find some mathematical symbols.

https://en.wikipedia.org/wiki/List_of_mathematical_symbols_by_subject