Module 1 Handout

Definitions:

A ***Problem (Computational)*** is a collection of mathematical or logical questions.

An ***Algorithm*** is a step-by-step procedure for solving a problem in a finite amount of time. (Note the finite amount of time is a required part of the definition)

An ***Implementation*** is a reduction of an algorithm to an executable format.

The ***Best Case*** is the input set that causes an implementation or algorithm to use the fewest resources.

An ***Average Case*** is the input set that causes an implementation or algorithm to use the typical resources. Average cases are more difficult to compute and typically assume a particular probabilistic distribution of the input.

The ***Worst Case*** is the input set that causes an implementation or algorithm to use the most resources.

For each of Best, Average and Worst case input sets we can write functions describing the resources needed based on the input size.

Asymptotic Bounds:

Big-Oh [such as $O(n^2)$] is a possibly tight upper bound on a function.

Little-Oh [such as $O(n^2)$] is an upper bound on a function that is not asymptotically tight

Big-Omega [such as $\Omega(n^2)$] is a possibly tight lower bound on a function.

Little-Omega [such as $\omega(n^2)$] is a lower bound on a function that is not asymptotically tight

Big-Theta [such as $\Theta(n^2)$] is a tight upper bound and lower bound on a function.

Notice that these bound FUNCTIONS and not algorithms! We use them for algorithms by first writing a function that relates the time or memory an algorithm requires to the size of the input. Then we bound THAT FUNCTION with one of the above bounds.