CMPT 225 - Big Oh

If f,g are functions f:N->N, g:N->N
f is O(g) means there are constants n0, c > 0
such that
for every n > n0, f(n) <= c*g(n)

The n0 means that there are finitely many small values that don't matter, we care about the end behaviour

Asymptotic Notation (eg. Big-Oh)

Is not about algorithms
Is a toold for describing (growth of) functions
It is useful for describing functions related to algorithms + data structures
eg - minimum or maximum time/space taken

We use it so often for worst-case time for an algorithm that we often leave implicit a statement like "let T(n) be the max time taken by algorithm A on an input of size at most n"
This statement is essential.