Lesson 1 - Computational Complexity

Types of Complexity:

1. Time Complexity
   1. The analysis of the overall amount of time or work required to solve a problem usually measured in FLOPS (Floating Point Operations)
   2. This is the number of steps a computer must take
   3. Not same as clock time
2. Space Complexity
   1. The analysis of overall storage (memory locations) required to solve a problem of a given size, usually measured in storage locations (bytes, floating points numbers integers, etc.)

Example 1:

Describe a basic algorithm for finding the maximum value of set of numbers

In the above example we see that as the size n of a set increase. The number of operations increases in a linear manner.

We say in this case that Algorithm 1 is “big-O of n”. This simply means that the number of operations required is no greater than directly proportional to n.

Definition of big-O

Let f ang be functions from the set of integers to the real number. We say that f(x) is O(g(x)) is there are constants C and K such that

|f(x)| <= C|g(x)|

Whenever x>K. We say that “f(x is big O of g(x) in this case”.

Algorithm 1 requires n-1 steps |n-1| <= |n+0| = 1|n|

We use big O notation to estimate the growth of a function without worrying about constant.

E.g.

100n2 +17 n +w id O(n2 )

N3 is O(n3 )