**Time complexity-** Time complexity is the time taken by an algorithm to complete a set of instructions. The time efficiency for an algo is very imp as it will show us the efficiency of our algorithm. Space complexity is also important as it tells how much space will be occupied to do some iteration.

The time complexity for different hardware is different as new technology hardware will have better speed of doing tasks and old ones will have outdated version of hardware and technology so we cannot compare the time taken by these two as it is going to have a very big difference. Whenever we run our code in the system it is going to take different time.

The time complexity is of three types, best, average and worst. We always find out the worst complexity for our algos so that we can say that this is the case where my algo performs worst otherwise it will perform average or best.

The complexity of a code depends a lot on the number of loops we have and how many times it is being iterated. For example if we have a loop which runs n times and it is giving values for a function n times then the time complexity is going to be O(n^2). If there is only one loop then the time complexity is O(n). For some problems we will need to iterate through the loop n number of times and find out the output for each iteration and it is going to give us the correct info about the time complexity as we solve the function that we will get. Sometimes it is going to be a log function as well, which has even less time complexity.