Time Complexity

1. The runtime of an algorithm can depend on several factors but here we consider only the input. We can say that we find time taken as a function of input.
2. We consider a model machine which has single processor, works on 32 bit arch and flow of instruction is sequential also it takes 1 unit of time for arithmetic, logical, assignment statements.
3. We see that many functions will behave in a similar way when the value of n tends to infinity. So, we use asymptotic notations (Big O notations).

What is Big O notation?

It just depicts the upper complexity of the algorithm, for e.g. O(n3 ) means that our algorithm can possibly get executed in linear time complexity but cannot exceed the complexity of the n3 .

TYPES OF NOTATIONS

What is a Big O?

 Big O stands for ‘order of’

Worst case🡪 big oh 🡪 O

Best case 🡪 big omega 🡪 Ω

Average case🡪 big theta 🡪 Θ

Steps to find Time Complexity of an Algorithm