1

Time complexity: The amount of time required to run the complete program in computer is called the time complexity.

2

- In Best case, the algorithm performs the minimum no. of steps on input data of n elements.
- Worst case is the function which performs the maximum no. of steps on input data of n elements.
- Average case is the function which performs ~~the minimum~~ an avg no. of steps on input data of n elements.

i) Big Oh Notation: If the function $f(n) = O(g(n))$, then there exists a +ve constants n_0, c such that $f(n) \leq c \cdot g(n)$ $\forall$ $n \geq n_0$, $c > 0$, $n_0 \geq 1$

Worst slow

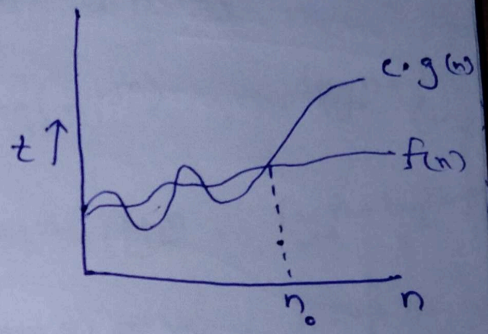
Ex: i)

$f(n) = 3n + 2 \quad g(n) = n$

$f(n) \leq c \cdot g(n)$

$3n + 2 \leq c \cdot n$

$n = 1$ ✗	~~$c = 0$~~ ✗
$n = 2$ ✓	$c = 1$ ✗
$n = 3$ ✓	$c = 2$ ✗
⋮ ✓	$c = 3$ ✗
	$c = 4$ ✓



$\Rightarrow T(3n+2) = O(n)$

if $n \geq 2$, $c = 4$

3

02/05/22

Amortized Analysis:

4

→ In an amortized analysis, we averge the time required to perform a sequence of datastructure operations over all the operations performed.

5

5.) If $f(n) = 5n^2 + 6n + 4$

To prove: $T(n) = O(n^2)$

In the above function, $5n^2$ is the term having the highest power

So, the ~~variable~~ function 'n^2' will be time complexity in worst case, by excluding constants.

29/04/22

Fibinacci Series:

0, 1, 1, 2, 3, 5, 8, 13, . . .

6

$$T(n) = \begin{cases} 0 & n=0 \\ 1 & n=1 \\ T(n-1)+T(n-2) & [\text{else}) \ n \geq 2] \end{cases}$$

$$T(n) = T(n-1) + T(n-2)$$

$$\Rightarrow T(n) = 2T(n-1) + C$$

$$= O(2^n) \quad (UB)$$

$$\Rightarrow T(n) = 2T(n-2) + C$$

$$= O(2^{n/2}) \quad (L.B)$$

}

7

Towers of Henoi

$T(n) = 2T(n-1) + c$ - Recurance eq.

$a = 2$
$b = 1$
$f(n) = c$

$O(2^n)$ - Time complexity

Efficiency looks at **how much time it takes to run a particular algorithm and how much space is needed**. By using

What are the activities or processes involved in evaluation?

In this section, each of the four phases is discussed.

9

- Planning. ...
- Implementation — Formative and Process Evaluation. ...
- Completion — Summative, Outcome, and Impact Evaluation. ...
- Dissemination and Reporting.

BASIC ASYMPTOTIC EFFICIENCY CLASSES

- Constant.
- log n. Logarithmic.
- n. Linear.
- n log n. n-log-n or linearithmic.
- n2. Quadratic.
- n3. Cubic.
- 2n. Exponential.
- n! factorial.

10

i) Size of input

ii) Software support

iii) Hardware environment.

18/04/22

12

Space Complexity:

How much additional space is required for an algorithm is space complexity.

v) n, $(\log n)^{100}$

$\log n$ $\log(\log n)^{100}$

$\log n$ $100 \log_2(\log_2 n)$

$\log_2 2^{1024}$ $100 \log(\log 1024)$

1024 $100 \log(\log 2^{10})$

$100 \log_2(10)$

$1024 > 1000$

$f(n) = n! + n\log n + n^{50}$

$O(n) = (n!)$

14

Order of growth in algorithm means how the time for computation increases when you increase the input size. It really matters when your input size is very large.