ADS LAB ASSIGNMENT 2

Name - Khushbu Jain

Roll Number - 23115047

Q.1 Computer scientists have defined several categories of algorithm efficiency as shown below:

Any measure of efficiency presumes that a sufficiently large sample is being considered. If you are dealing with only 10 elements & the time required is a fraction of a second, there is no meaningful difference between two algorithms. The table above depicts the estimate of the time needed to solve the problem given different efficiencies, for e.g., if we are using a computer that executes a million instructions per second & the loop contains 10 instructions, then 0.0001 second spent for each iteration ~~for~~ of the loop.

Sol →

Table : Measures of Efficiency for n = 10000.

Efficiency	Big-O	Iterations	Estimated Time	Best Case	Average Case	Name of related Algorithm
Logarithmic	$O(\log n)$	14	microsecond	$O(1)$	$O(\log n)$	Binary search
Linear	$O(n)$	10,000	seconds	$O(1)$	$O(n)$	Linear search or Traversal in Array
Linear Logarithmic	$O(n\log n)$	140,000	seconds	$O(n)$	$O(n\log n)$	Merge sort, Heap Sort.
Quadratic	$O(n^2)$	$10{,}000^2$	minutes	$O(n\log n)$	$O(n^2)$	Bubble Sort, Insertion Sort
Polynomial	$O(n^k)$	$10{,}000^k$	hours	$O(n^2)$	$O(n3)$	Matrix Multiplication, Floyd Warshall
Exponential	$O(c^n)$	$2^{10{,}000}$	Intractable	O(2^n)	O(2^n)	Subset Sum, Travelling Salesman
Factorial	$O(n!)$	10,000!	Intractable	$O(n!)$	$O(n!)$	Brute Force Permutations

Q.2 The definition for recursive factorial algorithm is

$$factorial(n) = \begin{cases} 1 & , n = 0 \\ n * factorial(n-1), & n > 0 \end{cases}$$

List down the recursive definition of atleast more than five different problems?

Sol →

1) Fibonacci sequence

$$fibonacci(n) = \begin{cases} 0 & , \text{if } (n == 0) \\ 1 & , \text{if } (n == 1) \\ fibonacci(n-1) + fibonacci(n-2), & \text{if } n > 1 \end{cases}$$

The fibonaaci sequence is defined by adding the previous two terms.

2) Sum of first n Natural Numbers

$$sum(n) = \begin{cases} 0 & , \text{if } n == 0 \\ n + sum(n-1) & , \text{if } n > 0 \end{cases}$$

3) Power function (x^n)

$$power(x, n) = \begin{cases} 1 & , \text{if } n == 0 \\ x * power(x, n-1) & , \text{if } n > 0 \end{cases}$$

4) GCD (Greatest Common Divisor)

$$gcd(a, b) = \begin{cases} a & , \text{if } b == 0 \\ gcd(b, a \% b) & , \text{if } b \neq 0 \end{cases}$$

5) Check palindrome String

$$isPalindrome(str, start, end) = \begin{cases} true & , \text{if } start >= end \\ false & , \text{if } str[start] != str[end] \\ isPalindrome(str, start+1, end-1) & , \text{otherwise} \end{cases}$$