Date: 04/23/12
Page No. 5

Name: Govind Sharma
Section: DS1
Roll No: 51

Assignment-01

Q1 Asymptotic notations are known as mathematical tools which are required to describe the limiting behaviour of a function as it's input tends towards infinity.

1) Big O notation (O):
This notation represents the upper bound of an algorithm's running time in the worst case scenario.
For Ex: $O(n^2)$ for a quadratic equation.

2) Omega notation:
This represents the lower bound of an algorithm's running time in the best case scenario.
Ex: $\Omega(n)$ for a linear algorithm.

3) Theta notation (Θ):
This represents both the upper bound as well as lower bound providing a tight bound on the algorithm's running time.
Ex:
$\Theta(n)$ for a linear algorithm.

Date :

Page No.

Q2 The time complexity of the given code is $O(\log n)$. Since the variable i doubles in each iteration.

Q3 The recurrence relation $T(n) = 3T(n-1)$ represent the expotential growth. The time complexity is $O(3^n)$.

Q4 The recurrence relation $T(n) = 2T(n-1) - 1$ represent exponential growth. time complexity is $O(2^n)$.

Q5. Time complexity : $O(n^{1/2})$

The loop is going to iterates until the sum 'S' exceeds "n" which happens when "i" reaches $(n^{1/2})$.

Q6 Time complexity : $O(n^{1/2})$ iterates when $i*i$ is greater than n, which will happen when i reaches $(n^{1/2})$.

Q7 Time complexity : $O(n \log n)$

Q8 The inner and outer loop will run n times, making it $O(n^2)$. Additionally function recursively call with $n-3$ so the no of times it recurses can be represented as $n/3$

$$O(n^2) * O(n/3) = O(n^3).$$

Date :

Page No.

Q9 The outer and inner loop will run n times.
Therefore time complexity of second function is

$O(n) * O(n/1 + n/2 + n/3 + \dots n/n$

hermonic series is $\log n$.

So time complexity is $O(n \log n)$.

Q10 n^K growth increases polynomially with n.
c^n growth rate increases exponentially with n.

c^n grows faster than
if $c>1$ & $k>0$ then

$$\lim_{n\to\infty} \frac{c^n}{n^x}$$

Therefore for any $c>1$ & $K>0$ c^n growes faster than n^K.