Algorithm Design and Algorithms

UID: U00790636

Q.4 Ans:

Initially begin with purely exponential or polynomial functions.

The best way to solve these kinds of problems is first let's start with exponential and polynomial functions.

Here, the function g3(n)= $n(\log n)^3$ Which can be written as $O(n^{4/3})$.

So the growth rate of this function is $n(\log n)^3 = O(n^{4/3})$. Which means g3 will follow g4.

Now let's consider polynomial functions, g2 and g7 both are polynomial functions. And the order it follows is g2 and g7.

Here $2^n = O(2^{n2})$ and

$2^{n2} = O(2^{2n})$.

Let's look at the functions which have involved both exponential and logarithms.

Comparing upon the g3 and g4 functions it is evident that the g3 will have less growth rate than g4.

Let's say the base algorithm for all the logs here in the question is 2 and the function g1 returns the same value as n ½ which becomes the regular equation. So now this can written in according to growth order is $2^{\sqrt{\log n}} = O(2\log n)$.

So we can conclude that √logn is followed by logn.

$n^{4/3} < O(n^{\log n})$ this is because the n power is always some part of n(power)logn.

Other remaining functions are g6, g7, g2. Let's compare g5 with these three functions.

G5 which is n(log n) and g6, g7 and g2 are exponential function of 2, which follows the below order.

N(log n) will have less growth rate, 2(n) and 2(n)2 will follow one by one. At the end the g7will have highest growth order.

2√log n and (n (log n)) = $2\sqrt{}^{\log n} = 2^{(\log n)1/2}$ which can be written as $O(2^n)$

Growth rate for exponential will be always more so that these functions fall at last in the sequence when comparing with the growth rate.

The following is the ascending order of the growth rate functions:

$g_1 < g_3 < g_4 < g_5 < g_2 < g_7 < g_6.$