Name - Rishee Bartwal
Rollno - 16
CE - SPL

HITAISHI
Date

DAA Assignment - 1

1) Asymptotic notation is used to describe the growth rate of a function as the input size increases to infinity. Its commonly used in computer science and algo analysis to decribe its performance

There are 3 different type of asymptotic notation:- big O, big theta (Θ) and big omega (Ω)

a) Big O notation - It describe worst case running time of a program
b) Big Ω Notation - Describes the best running time of a program
c) Big-Θ notation - It is computed by counting no. of iterations the algo always takes with an input n

2) Time complexity is O(logn)
3) $T(n) = 3\,T(n-1)$ if $n > 0$, otherwise1

Substitute $T(n-1)$ be first eq

$$T(n) = 3[3\,T(n-2)] = 3^2\,T(n-2)$$
$$T(n) = 3^k\,T(n-k)$$

Repeating untit we reach base case $n = 0$
let $k = n$ then

$$T(n) = 3^n\,T(0)$$

as $T(0)$ = constant

$$T(n) = O(3^n)$$

Therefore time complexity is $O(3^n)$

(4) Substitute $T(n-1)$.

$$T(n) = 2[2T(n-2) - 1] - 1 = 2^2 T(n-2) - 2 - 1$$
$$= 2^2 T(n-2) - 3$$

$$T(n) = 2^k T(n-k) - (2^k - 1)$$

repeating this until base case $n = 0, k = n$

$$T(n) = 2^n T(0) - (2^n - 1)$$

$$T(n) = O(2^n)$$

Hence time complexity is $O(2^n)$

(5) Time complexity is $O(\sqrt{n})$

(6) Time complexity is $O(\sqrt{n})$

(7) Time complexity is $O(n \log n)$

(8) Time complexity is $O(n^5/27)$

(9) Time complexity is $O(n \log n)$

(10) The asymptotic relation between n^k and c^n is determined by the limits of their ratios as n approach ∞. And for the fun n^k grows at polynomial rate while c^n grows at exponential rate. In which exponential is faster

$$\lim_{n \to \infty} c^n / n^k$$

Using L Hospital's rule

$$\lim_{n \to \infty} \left[(\ln c)^n / (k\, n^{(k-1)}) \right]$$

WKT $f(n) = O(g(n))$

if there exist constant c & n0 such that for all $n \geq n_0$.

$$|f(n)| \leq c\,|g(n)|$$

We can say c^n is $O(n^k)$ if there exist constant c & no such that for all $n \geq n_0$, $|c^n| \leq c|n^k|$

Taking b/s log

$$n \log(c) \leq k \log(n) + \log(c)$$

Divide b/s by log(n)

$$\log_c(n) \leq k + \log_c(\log(n)) - \log_c(c)$$

let $x = \log_c(n)$

$$k + \log_c(\log_c x) - 1 = 1$$

By solving

$$x = \log_c(c^{2-k}) = 2 - k$$

Therefore we get c^n is $O(n^k)$ for all $n \geq n_0$ by setting $c = e$ & $n_0 = e^{(2-k)}$