DATA STRUCTURE

NAME: AMIT MOHANPURIA
Rol No.: 11912083
Branch: I.To

ASSIGNMENT : 1

Q.1.) $t_{n=10} = 5\,sec.$

$t_{n=5} = ?$

$$Kn^2 = t$$

$$K \times 100 = 5$$

$$K = \frac{1}{20}$$

$$t_{n=50} = \frac{1}{20} \times 50 \times 50$$

$$= 125\,sec.$$

Q2.) $T_A = n^3$ $\qquad T_B = 2n^2$

break pt.

$$n^3 = 2n^2$$

$$n = 2$$

Q3.) $f(n) = n2^n$, $g(n) = 4^n$

Applying limit rule,

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \to \lim_{n\to\infty} \frac{n2^n}{4^n} \Rightarrow \lim_{n\to\infty} \frac{n2^n}{2^{2n}}$$

Applying L'Hospital $\Rightarrow \lim_{n\to\infty} \frac{1}{2^n \log 2}$

Hence, $f(n)$ is $O(g(n))$

Q4.) Let polyno. f^n $P(n)$ and $\log(n)$ f^n,

using limit rule, $\lim_{n\to\infty} \frac{\log(n)}{P(n)}$ $\left[\frac{\infty}{\infty}\right]$

Applying L'Hospo rule,

$$\Rightarrow \lim_{n\to\infty} \frac{1}{n(P_n)}$$

$$\Rightarrow 0$$

thus $\log n$ grows slower than all f^n

Let 2 $\log$ f^n $\dfrac{\log_a n}{\log_b (n+1)}$ and.

Applying lint rule,

$$\lim_{n \to \infty} \frac{\log n}{\log(n+1)} = \lim_{n \to \infty} \frac{\log n}{\log b + \log(1 + \frac{1}{n})}$$

$$= \lim_{n \to \infty} \frac{\log_a n}{\log_b b}$$

$$= \lim_{n \to \infty} \frac{\ln b}{\ln a}$$

Hence all log. fun'' grow rate, &

Oe, constant.

Q5.)a.) Average case (θ)	Worest case (0)
* Performs average no. of steps per input data of n elements.	* Performs maximum no. of steps on input data. of n elements
* ~~Averaged~~	* Input is arbitrary size
* Averaged over all poss-ible inputs.	& output & upper bound
eg:- $\theta(n \log(n)) \to$ Quicksort.	eg: $O(n^2) \to$ Quick short.

Q5) b.)

Worest Case (0)	Asymptotically Bounded (Ω)
* I/P is arbeitary &t c & O/P is upper bound.	* Growth of a running time to within const. factors below.

Q6.)

a.) $f(n) = n^4 + \log n + 12$, $g(n) \cdot n^4$

$$\lim_{n \to \infty} f(n) \Rightarrow \lim_{n \to \infty} \frac{n^4 + \log + 12}{n^4}$$

$$\Rightarrow \lim_{n \to \infty} \frac{4n^3 + \frac{1}{n}}{4n^3}$$

$$\Rightarrow \lim_{n \to \infty} \frac{4n^3}{4n^3} + \frac{1}{4n^4} \nearrow^0$$

$$\Rightarrow \perp \text{ const.}$$

Hence $f(n)$ is $O(g(n))$

Q.1)

a.)

$K = 1$

$K = 2$

$K = 3$

$\vdots$

$O(n-1)$

$\Rightarrow \in O(n)$

$K = n-1$

b.)

for $i = 1$ $j = 2, 3 \ldots \ldots n$ $(n-1)$

$i = 2$ $j = 3, 4 \ldots \ldots n$ $(n-2)$

$i = n-1$ n $(n - i)$

$n - \{(n-1)+1\}$

$(n-1) + (n-2) + (n-3) + (n-4) \ldots \ldots 3 + 2 + 1$

$$\frac{n(n-1)}{2}$$

$O(n)$

Q.2)

Quadratic Function $\Rightarrow f(n) = n^2$

$n * n = n^2$

$1 + 2 + 3 \ldots \ldots (n-1) + n = n * (n+1)/2$

Quadratic algo. are practical for relatively small problems, whenever n doubls, the running time increass forefold

$1, 4, 16, 64, 256, 1024, 4096$

Q9) $T_A = 100^n$

$T_B = n^4$

Using limit rule, $\lim \dfrac{n^4}{100^n}$ $\quad \left[\dfrac{\infty}{\infty}\right]$

$\lim\limits_{n\to\infty} \dfrac{4n^3}{100^n \log 100}$

$\lim\limits_{n\to\infty} \dfrac{24}{100^n (\log(100))^4} = 0$

Hence, T_A grows faster when $n\to\infty$

Q10) $f(n) = n\log n$ $\qquad g(n) = \log(n!)$

$\qquad = \log(1.2.3.4 \cdots (n-1)n)$

$\qquad \le \log\left(\dfrac{n}{2}^{n/2}\right)$

$\qquad = \dfrac{n}{2}\log\dfrac{n}{2}$

$\Rightarrow \dfrac{n}{2}\log n - \dfrac{n}{2}\log 2$

$\Rightarrow \dfrac{n}{2}\log n - \log 2) \simeq \Theta(n\log n)$

Hence, $f(n) \in \Theta\, g(n)$

Q(1.) a.) $2^{n+1} + 4^{n+1}$

for Θ $c_1 \, g(n) \leq f(n) \leq c_2 \, g(n)$

$2^{n+1} + 2^{2n+2}$

$\Rightarrow \dfrac{2^n}{2} + 4 \cdot 2^{2n}$

$\Rightarrow 2^{2n}\left(4 + \dfrac{1}{2^{n+1}}\right)$

Higest order term 2^n, $\rightarrow \Theta(4^n)$

b.) $(n^2 + 6)^8$

higest order term $= (n^2)^8$

$= n^{16}$

$\Rightarrow \Theta(n^{16})$

12.) $T_A = n^2$

$T_B = n+1$

Breaking Pt.

$n^2 = n+2$

$n^2 - n - 2 = 0$

$n^2 - 2n + n - 2 = 0$

$n(n-2) + 1(n-2) = 0$

$\boxed{n = 2}$