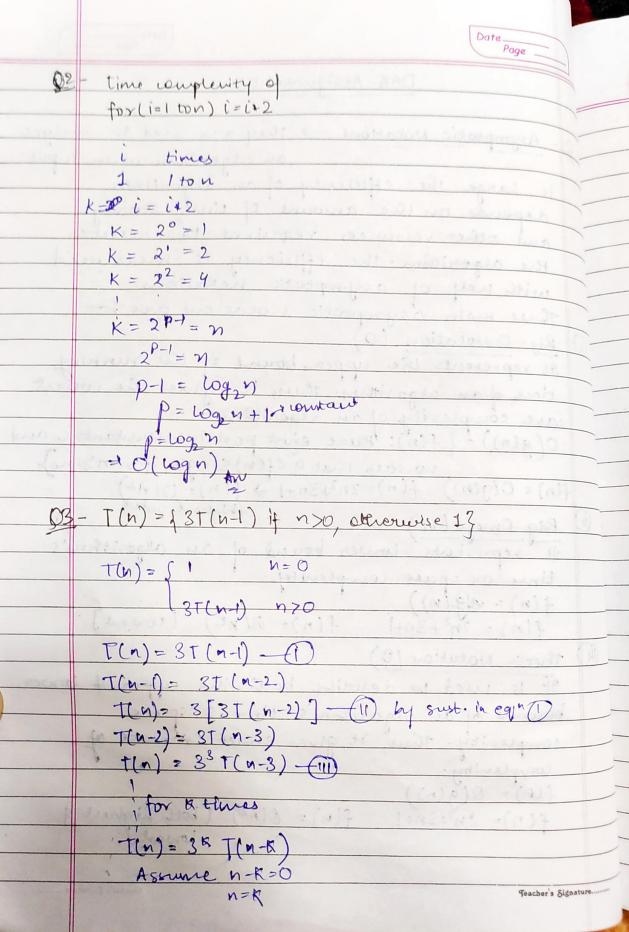
| Date | | 1 |
|------|------|---|
| | Page | - |

DAA-Assignment

DI - Asymptotic Notations -> they are used to analyze an algorithm when Input ic large. the efficiency of an algoritam depends on the amount of time, storage and other resources required to energite the algorithm. the efficiency is measured with holp of asymptotic netations. Three main asymptotic notations are; -Big- O Notation (0 It represents the upper bound of the running time of an algorithm. thus, it gives the worst case complexity of an algorithm. O(g(n)) = {f(n): Here exist positive constants c and f(n) = O(g(n)) f(n) = $2n^2+3n+1 \Rightarrow f(n) = O(n^2)$ Rea Outside (a) Big Omega(n) It represents lower bound of an algorithm's time or space complexity. f(n) = a(q(n)) f(n)= N(n2) (lowest] $f(n) = 2n^2 + 3n + 1$ Thera Notation (0) It is used to describe both the upper of lower bounds of an algorithm's time of space complexity. Thus, it gives exact extinate of complenity. f(n) = 0(g(n)) f(n) = O(n2) (both highert s f(n)= 2n2+3n+1



Date_____ Page ____

$$T(n) = 3^{n} T(0)$$

 $T(n) = 3^{n} (i)$
 $T(n) = 3^{n}$
 $T(n) = 3^{n}$

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

suba. $T(n-1)$ in eq. (1)
 $T(n) = 2(2T(n-2)-1)-1$

$$t(n) = 2(2t(n-2)-1)-1$$

$$t(n) = 2^{2}t(n-2)-2+-(1)$$

$$T(n) = 2^{2}(2T(n-3)-1)-2+$$

 $T(n) = 2^{3}T(n-3)-2^{2}-2-1$

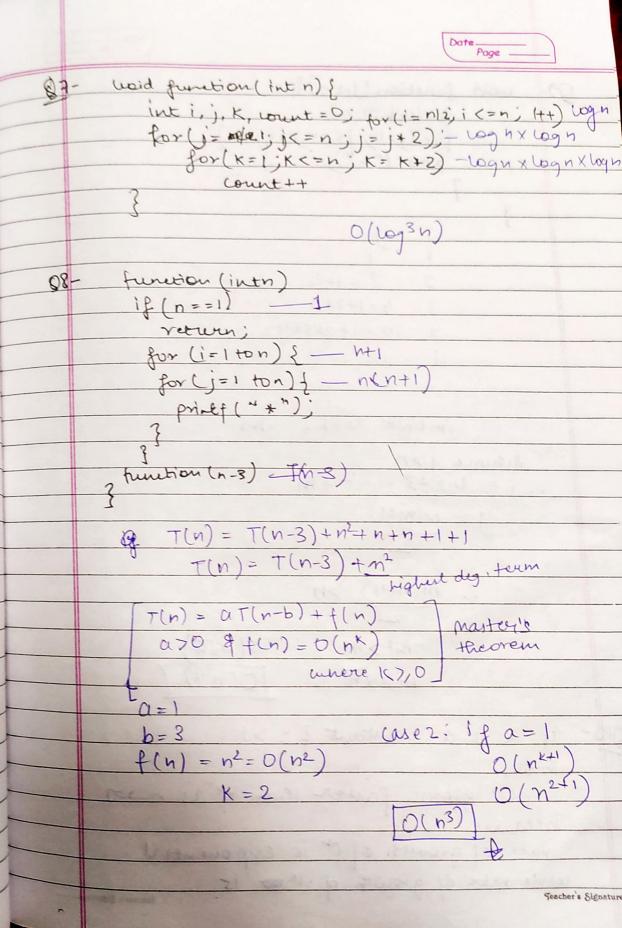
$$T(n) = 2^{n} T(0) - 2^{n+1} = 2^{n-2} = 2^{n-3} = --- 2^{2} - 2^{1} - 2^{0}$$

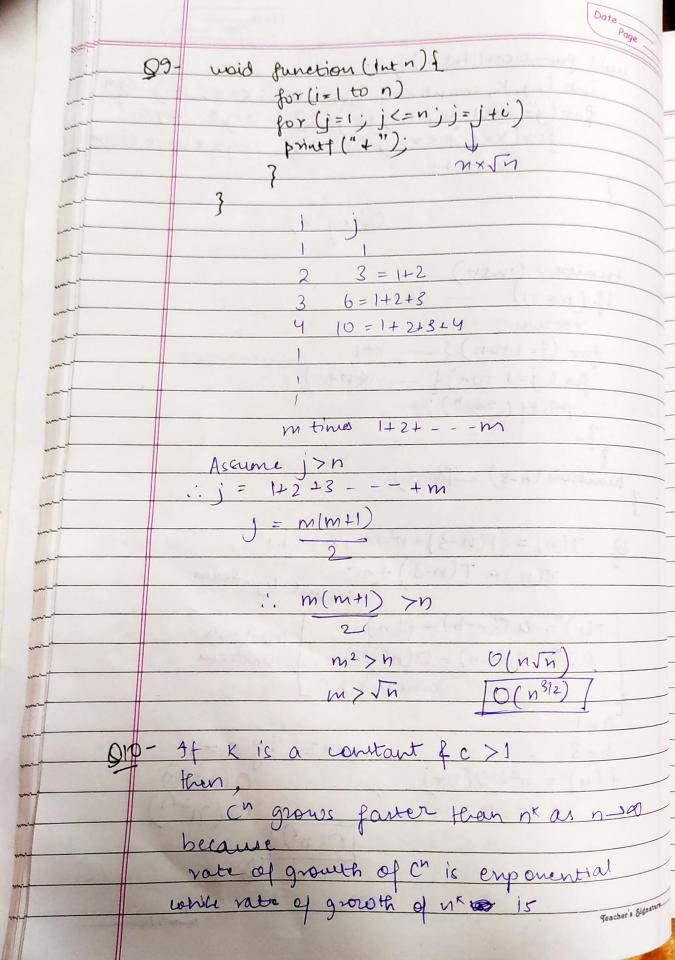
 $\pm 2^{n} - 2^{n+1} - 2^{n-2} - 2^{n-3} = --- 2^{2} - 2^{-1}$

$$\Rightarrow 2^n - (1(2^{n-1}))$$

$$32x - 2x + 1$$
 $S = \alpha(x^n - 1) > 1$

Q5- int i=1; s=1; while (S(=n){ i++; S=Sti; print (" *"); while (ix n) 2 1+2 =3 3 1+2+3 = 6 4 1+2+3+4=10 Assume s>n :, s=(1+2+3-m) → m (m+1) ni times 1+2+3---+m (1, m(m+1) > nm>Jm 10(vn) 86 - wold function (int n) { int i, count=0; for (i=1; i+i <=n; i++) count ++; 1 x i > n i+i=K2 k2> n K>1/n Teacher's Signature.





Date Page ___ prolynomial.

Or we son cay ("= O(NK).