**PART 2   
(This should take 2.5 hours by one team)**

**Solve all the following recurrences.**

1. **T(n) = T(n-1) + n**

**cost**

**T(n)        ----->      (n)**

**T(n-1)     ----->       (n-1)**

**T(n-2)   ----->   (n-2)**

**….**

**T(1)------>            (1)**

**timeComplexity:  n + n-1 + n-2+ ….+1= O(n2)**

**1+2+3+....+n-2+n-1+n=  n(terms)\*n-->n2**

1. **T(n) = T(n-1) +(n)2**

**cost**

**T(N)     ----> (N)2**

**T(N-1) ---> (N-1)2**

**T(N-2) ---> (N-2)2**

**….**

**T(1)    --->   (1)2**

**timeComplexity:  (n)2 + (n-1)2 + (n-2)2+ ….+(1)2=O(N3)**

**(1)2 + (2)2 + (3)2 +....+ (n-2)2 +(n-1)2 +(n)2 =**

**N(terms) \* N2   = N3**

1. **T(n) = T(n/2) +(n)2**

**1st                   T(n) ---> (n)2**

**2nd                 T(n/2) ---> (n/2)2**

**T(n/4) ---> (n/4)2**

**….**

**T(1) ----->(1)2**

**(N)2 + (N/2)2 + (N/4)2 +N/8 +.....+32+16+ 8 +4 +2+1=Second MASTER THEOREM APPLIED= O(N2)**

1. **T(n) = T(n/2) +n3**

**cost**

**T(n)        ----->      (n)3**

**T(n/2)     ----->    (n/2)3**

**T(n/4)     ----->    (n/4)3**

**……..**

**T(1) ------>         (1)3**

**(N)3 + (N/2)3 + (N/4)3 + (N/8)3 +.....+ (16)3+ (8)3 +(4)3 +(2)3+(1)3=**

**Second MASTER THEOREM APPLIED = O(N3)**

1. **T(n) = 4T(n/2)+n2**

**cost**

**T(n) ----->                                            (n)2**

**Call                   Cost**

**T(n/2)   T(n/2)   T(n/2)  T(n/2) ---->   (n/2)2 + (n/2)2 + (n/2)2 + (n/2)2 ⇒**

**(n2/4) + (n2/4) + (n2/4) + (n2/4) --->  (n2/2) +  (n2/2) ⇒ (n)2**

**Call: T(n/4)T(n/4)T(n/4)T(n/4)  T(n/4)T(n/4)T(n/4)T(n/4)   T(n/4)T(n/4)T(n/4)T(n/4)  T(n/4)T(n/4)T(n/4)T(n/4)**

**Cost: (n/4)2+(n/4)2+(n/4)2+(n/4)2 + (n/4)2+(n/4)2+(n/4)2+(n/4)2 + (n/4)2+(n/4)2+(n/4)2+(n/4)2 + (n/4)2+(n/4)2+(n/4)2+(n/4)2**

**(n2/16)+(n2/16)+(n2/16)+(n2/16) + (n2/16)+(n2/16)+(n2/16)+(n2/16) + (n2/16)+(n2/16)+(n2/16)+(n2/16)+ (n2/16)+(n2/16)+(n2/16)+(n2/16**

**4(n2/16)                      +   4(n2/16)                        +                   4(n2/16)             +        4(n2/16)**

**N2/4                           +              N2/4                        +     N2/4                 +         N2/4  ⇒  (n)2**

**for(int i=0;i<n;i\*=2) -----> logn times for(int i=n;i>0; i/=2) -------> logn times**

**1st master theorem  = Cost \* number of levels = N2 \*  logn ----> O(N2logn)**

1. **T(n) = 4T(n/2)+n3 COST**

**T(n) ----->                           (n)3    0th LEVEL               n3**

**Call Cost**

**T(n/2)   T(n/2)   T(n/2)  T(n/2) ---->   (n/2)3 + (n/2)3 + (n/2)3 + (n/2)3    ⇒    4(n3/8) -> n3/2 1st level n3/2**

**2nd Master theorem applied = 1st level cost = O(n3)**

1. **T(n) = 27T(n/3) + n3**

**T(n)   ----->        (n)3                           0th level               n3**

**27 Calls Cost of 27 calls**

**T(n/3) +T(n/3)+T(n/3)+.....+T(n/3)      (n/3)3 +  (n/3)3 +(n/3)3 +..... +(n/3)3  = 27(n3/27)   --->  1st level n3                  1st  Master Theorem applied=O(n3) \* Number of levels =n3 \* log3n=n3log3n -------> O(N3 logN)**

**t**

1. **T(n) = 27T(n/4) + n3**

**T(n)       --->          (n)3                           0th level                                           n3**

**27 Calls Cost of 27 calls**

**T(n/4) +T(n/4)+T(n/4)+.....+T(n/4)   (n/4)3 +  (n/4)3 +(n/4)3 +..... +(n/4)3  = 27(n3/64)  ---> 1st level O(n3)**

**2nd Master Theorem applied=1st level cost = O(n3)**

**27(n3/64) < n3 <64(n3/27)**

1. **T(n) = 3T(n/3) + n**

**T(n)   ----->        (n)                           0th level               O(n)**

**3 Calls Cost of 3 calls**

**T(n/3) + T(n/3) + T(n/3)       (n/3) + (n/3) + (n/3) 3(N/3)  →  1st level O(n)**

**(N/3)/3  → (N/3)\*(1/ 3) → N/(3\*3) ⇒ N/9**

**1st Master Theorem applied = Cost of each level (N)  \*  Number of levels (log3 N) ⇒ O(NlogN)**

1. **T(n) = 3T(n/3) + n2**

**T(n)   ----->        (n)                   0th level               n2**

**T(n/3) + T(n/3) + T(n/3)       (n/3)2 + (n/3)2 + (n/3)2→ 3(n2/9)->                 n2/3**

**2nd Master Theorem applied = 1st level cost ⇒  O(n2)**

1. **T(n) = 2T(n/2) + 1**

**T(n)       --->          (1)                           0th level                                    1**

**2 Calls       Cost of 2 calls**

**T(n/2) T(n/2)    (1) +  (1)   =                                                                      2**

**T(n/4) T(n/4)  T(n/4) T(n/4)         1 +1+1+1  =                                                                4**

**1+ 2+4 +8 +....+n   <2n ---> O(n)**

**3rd Master Theorem =last level ki cost=     2log2n → nlog22   -> n ->O(n)**

**log22= 1 log33=1 log44=1**

1. **T(n) = 2T(n-1) + 1**

**T(n) ----> (1) 20 =1 0th level**

**T(n-1) + T(n-1) ----> (1) + (1) = 2   21 --->2 1st level**

**T(n-2) + T(n-2) + T(n-2) + T(n-2)  ----> (1) + (1) + (1) + (1) =4 22 2nd level**

**20+21+22+.....+2n <= 2.2n   O(2n+1)--> O(2n)**

**1+2+ 4+ 8 + ….+ N <= 2N**

**3rd master theorem =last level cost = O(2n)**

1. **T(n) = 2T(n/4) + 1**

**T(n)       --->          (1)                               0th level               1**

**T(n/4) +T(n/4)  → 1+1 =2                            first level            2**

**4 Calls**

**T(n/4) +T(n/4) + T(n/4) + T(n/4) 1+1+1+1=4        2nd Level           4**

**3rd last  Master Theorem applied=last level cost=2log4n   → nlog42 → O(nlog2)**

1. **T(n) = 4T(n/2)+ n2 log n**

**T(n)  ------> N2 logN 0th level              N2 logN**

**Log(N/2) ⇒  LogN - log 2**

**(n/2)2logN + (n/2)2logN + (n/2)2logN + (n/2)2logN**

**⇒ (n2/4)logN + (n2/4) logN + (n2/4) logN + (n2/4) logN = ~~4~~ ( (n2 \* LogN)/~~4~~ ⇒ (N2 logN)**

**Cost remains the same, hence applying 1st Master theorem.**

**1st Master Theorem applied = Cost of each level (N2log2N)  \*  Number of levels (log2N)     ⇒    O(N2 log2N)**

1. **T(n) = 3T (n/2)+ n**

**T(n) ----> (n) 0th level               N**

**call**

**(n/2) + (n/2) + (n/2) ----> 3(n/2) = 3n/2 1st level               3N/2**

**Cost decreasing, hence applying 3rd theorem.**

**3rd Master theorem applied =last level cost= 3log2n---->(swap n & 3(log property)) nlog23  ⇒  O(nlog3)**

**T(N)  = a T(N/b) + Nc      ⇒  (alogbN ) Last level cost**

1. **T (n) = 3T (n/3)+ n/2                                                                      cost**

**T(n) --->                                                                                                      n/2**

**T(n/3) T(n/3) T(n/3)  --->    (n/6)+(n/6)+(n/6) =~~3~~(n/~~6~~)                                n/2**

**n/3 / 2 → n/3 \* ½ → n/6**

**1st Master Theorem applied = Cost of each level (N/2)  \*  Number of levels (log3n)    ⇒ n/2\*log3n → O(nlogn)**

1. **T(n) = n1/2 T(n1/2) + n**

**T(n) --->                                          n**

**T(n1/2) …...calls-> T(n1/2) ⇒ n1/2 calls -->n1/2+n1/2…..  =   n1/2(n1/2) →                   n**

**1st Master Theorem applied = Cost of each level (N)  \*  Number of levels ()     ⇒  (N(log(logn)))**

**NUMBER OF LEVELS**

**N1         N1/2     N1/4      N1/8+.......+**

N(½)^0, N(½)^1, N(½)^2, N(1/2)^3…, N(½)^K

N1/2^k = c

TAKING LOG ON BOTH SIDES

log N1/2^k = log c1 c= log c1

½k  log N = c cross multiply

1Log N = c2k

  TAKING LOG ON BOTH SIDES

    Log (log N) = log C2

Log( log N) =  log (C2K)

log(log N)= logC + log2k  ; logc=will give you a constant and we will ignore the constant in next step

    log(log N) =klog22   ; we know that log22 is 1

Log (log N) = k

1. **T(n) = 3T(n-1) + 1**

**-**

**T(n) --->                                                                                                      1**

**T(n-1) T(n-1) T(n-1)  --->     1+1 + 1 = 3                                                            3**

**T(n-2) + T(n-2)+T(n-2) + T(n-2) + T(n-2)+T(n-2)  + T(n-2)+T(n-2)+T(n-2) ----> (1) + (1) + (1) + (1) + (1) + (1) + (1) + (1) + (1)  --->     9 2nd level**

**3rd Master Therom = last level cost = 30+31+32+.....+3n <= 3.3n   O(3n+1)--> O(3n)**

**1+2+ 4+ 8 + ….+ N <= 3N**

**3rd master theorem =last level cost = O(3n)**

**19. T (n) = 3T (n/3)+ n1/2 cost**

**T(N) -------> n1/2 0th level**

**T(n/3) + T(n/3) + T(n/3)      -------> 3(n/3)½       =  3(n1/2/31/2)**

**(3n1/2)/31/2 ---> (3/31/2 ) \*  n1/2   → 31-½.n1/2 → 31/2n1/2 = (3n)1/2 1st level**

**Applying 3rd Master Theorem = 3log3n---->(swap n & 3(log property)) nlog33  ⇒ N  ⇒  O(N)**

**20. T(n) = 2T(n/4) + n1/2**

**T(n)  --->                        n1/2**

**T(n/4)  T(n/4)  --->  (n/4)½+(n/4)½ =~~2~~(n1/2/~~2~~) --->                                     n1/2**

**1st Master Theorem applied = Cost of each level (N1/2)  \*  Number of levels (logN)  = O( N1/2 logN)**

**21.  T(n) = 7T(n/3) + n2**

**T(N)   ---> n2**

**T(N/3) T(N/3) T(N/3) T(N/3) T(N/3) T(N/3) T(N/3) → 7(n/3)2 → 7(n2/9)                   7n2/9**

**Cost is decreasing**

**2nd Masterom applied =1st level cost = O(n2)**

**22.  T(n) = 7T(n/2) + n2**

**T(N)   ---> n2**

**T(N/2) T(N/2) T(N/2) T(N/2) T(N/2) T(N/2) T(N/2) → 7(n/2)2 → 7(n2/4)                   7n2/4**

**Cost is increasing(multiplication factor is big)**

**3rd Master theorem=last level cost=7log2n -> (swap 7 & n)  Nlog37  ⇒   O(Nlog7)**

**23.  T(n) = 2T(n/4) + n2**

**T(n)...........           n^2**

**(n/2) + (n/2) ----> 2(n/2)^2 = 2(n2/4)=                     n^2/2**

**2nd Masterom applied =1st level cost = O(n^2)**

**24.  T(n) = T(n-4) + n2**

**T(n) =          (n)2**

**T(n-4) =   ` ` (n-4)2**

**T(n-8) =    (n-8)2**

**Cost is not same also and it is not decreasing or increasing geometrically so no master theorem is applied we have solved by making is series as shown below**

**(n)2+(n-4)2+(n-8)2+ (n-12)2 +.....+(12)2+(8)2+(4)2+(1)2**

**(1)2+(4)2+(8)2+(12)2+....+(n-12)2+(n-8)2+(n-4)2+(n)2= N(terms) \* N2 → O(N3)**

**25.  T(n) = 2T(n-4) + n2**

**T(n)...........           1(n2)           ---->   20 (n)2**

**T(n-4)+T(n-4) = 2(n-4)2        -----> 2(n-4)2        --->     21(n-4)2**

**T(n-8) T(n-8) T(n-8)+T(n-8)  ----->         4(n-8)2       ---->      22(n-8)2**

**23(N-12)2**

**3rd Master theorem applied=last level cost=O(2n/4)**

**hint: for(int i=0;i<n;i+=4) -->n/4 terms       for(int i=n;i>0;i-=4)   ---> n/4 terms**

**26.  T(n) = T(n-1) + 1/n (solve this after watching sarfraz video)**

**//** [**Read what is Harmonic Series**](http://www.mathcaptain.com/calculus/harmonic-series.html)

**T(n)           1/n**

**T(n-1)       1/n-1**

**T(n-2)      1/n-2**

**1/n + 1/n-1 + 1/n-2 +...+⅓+½ +1/1 = harmonic series O(logn)**

**27.  T(n) = T(n-1)  + log n. cost**

**T(n) logn**

**T(n-1) (log(n-1))**

**T(n-2) (log(n-2))**

**Logn + (logn-1) + (logn-2) +....+log3+log2+log1~~= logn \* n terms ⇒  O(nlogn)~~**

**loga + logb = log(a\*b)**

**log(n \* (n-1) \* (n-2) \* (n-3) \* (n-4)\*.......1)**

**Log n!**

**28.  T(n) = T(n/2) + T(n/4) + T(n/8) + 1                 cost**

**T(n)  ---->                               1      (3)0**

**T(n/2) + T(n/4) + T(n/8)               1+1+1 =    3    (3)1**

**T(n/4) + T(N/8) + T(n/16)               9**

**3 Master theorem applied = last level cost= 3log2n  → nlog23  ----> O(nlog3)**

**depts⇒ 3log8n<d<3log2n   ⇒   Nlog83 < d < Nlog23**

**29.  T(n) = T(n/2) + T(n/4) + T(n/8) + (n)  cost**

**T(n) n**

**T(n/2) + T(n/4) + T(n/8)          (n/2)+(n/4)+ (n/8)                          7n/8  
  
                                                             Cost is decreasing(dividing factor is bigger)**

**2nd Master Theorem applied= 1st level cost= O(n)**

**N + (⅞)1 n+ (⅞)2 n+(⅞)3 n+....+**

**T(n)  = 3T(n/2) + n**

**T(n/2 ) = t(n/4) + t(n/4) + t(n/4)  + (n/2) \*3**

**30.  T(n) = T(n/3)+T(2n/3) + n cost**

**T(n)       n**

**T(n/3) T(2n/3)        (n/3) + (2n/3) → ~~3~~n/~~3~~ n**

**1st Master Theorem applied = Cost of each level (N)  \*  Number of levels= N\*logn ⇒  O(nlogn)**

**Number of levels= log3n**

**Number of levels= log3(2n) →(log plus property applied) log3n + log32(constant will be ignored) ⇒ log3n**

**31.  T(n) = log n x T(n/log n) + 2n cost**

**T(n) 2n**

**T(n/logn) \* logn calls ------> cost calculation: ~~Logn~~(2(n/~~logn~~))                            2n**

**1st Master Theorem applied = Cost of each level (2n)  \*  Number of levels= 2N \* loglognn  =O(Nloglognn)**

**loglognn**

**32.  T (n) = 6T (n/3)+ n2 log n cost**

**T(n) n2logn**

**T(n/3)T(n/3)T(n/3)T(n/3)T(n/3)T(n/3)       6((n/3)2log(n)) → ~~6~~(n2/~~9~~)logn              (2n2/3)logn**

**6T(n/9) ---> 6((n/9)^2log(n) --->  6(n^2/81)logn  →      6/81 (n^2logn)**

**logn/3= log3n - log33 → log3n - 1**

**2nd Master theorem =1st level cost= O(n2logn)**

**33.  T (n) = T(n/4) + 5T(n/5) + T(n/3) + n3**

**T(n) (n)3**

**T(n/4) + T(n/5)+...+T(n/5) + T(n/3)     costs →  (n/4)3 + 5(n/5)3 + (n/3)3 →**

**(n3/64)+~~5~~(n3/~~125~~)+(n3/27) ⇒ 4003n3/43200    4003n3/43200**

**2nd Master Theorem applied = 1st level cost =O(n3)**

**34.  T (n) = T(n/4) + 5T(n/5) + T(n/3) + n2**

**T(n) n2**

**T(n/4) + T(n/5)+...+T(n/5) + T(n/3)     costs →  (n/4)2 + 5(n/5)2 + (n/3)2 →**

**(n2/16)+~~5~~(n2/~~25~~)+(n2/9) ⇒ (n2/16)+(n2/5)+(n2/9) ⇒ 269n2/720             269n2/720**

**2nd Master Theorem applied = 1st level cost =O(n2)**

**35.  T (n) = T(n/4) + 5T(n/5) + T(n/3) + n**

**T(n) n**

**T(n/4) + T(n/5)+...+T(n/5) + T(n/3)     costs →  (n/4) + ~~5~~(n/~~5~~) + (n/3) →**

**(n/4)+(n)+(n/3) ⇒19n/12                                                                                      19n/12**

**depths= log4n**

**depths = log5n**

**depths= log3n  ( this will be the max depth)**

**Log5n < log4n <log3n**

**3rd Master Theorem applied = last level cost =7log3N   → Nlog37 → O(nlog 7)**

**36.  T (n) = T(n/4) + 5T(n/5) + T(n/3) + 1           cost**

**T(n)  1**

**T(n/4) + 5T(n/5) + T(n/3)    1+5(1)+1 → 1+5+1 → 7**

**depths= log4n**

**depths = log5n**

**depths= log3n  ( this will be the max depth)**

**3rd Master Theorem = last level cost=7log3n → nlog37   → O(nlog7 )**

**37.  T(N) = T(n1/2)+ N**

**T(N) N  N**

**T(N1/2)                 N1/2 N/2**

**T(N1/4)           N1/4 N/4**

**N(½)0         N(½)1     N(½)2      N(½)3+.......+ N(½)k**

   N + (N)(½)^1

N + N/2 +N/4 +.....+ 4+2+1 <= 2n

2nd master theorem =1st level cost = O(N)