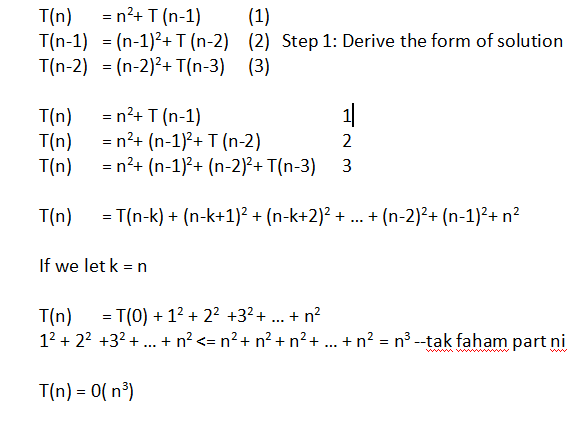
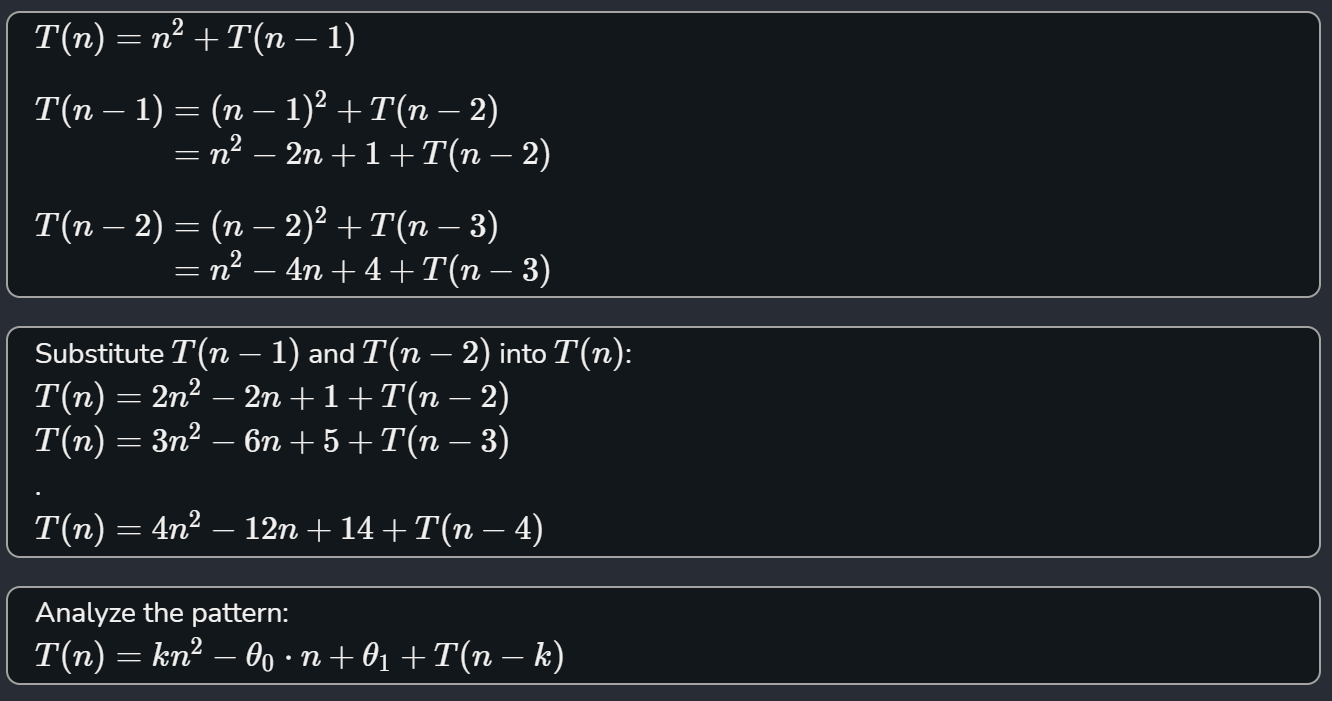
**WIA2005 Algorithm Design & Analysis**   
 **Semester 2**

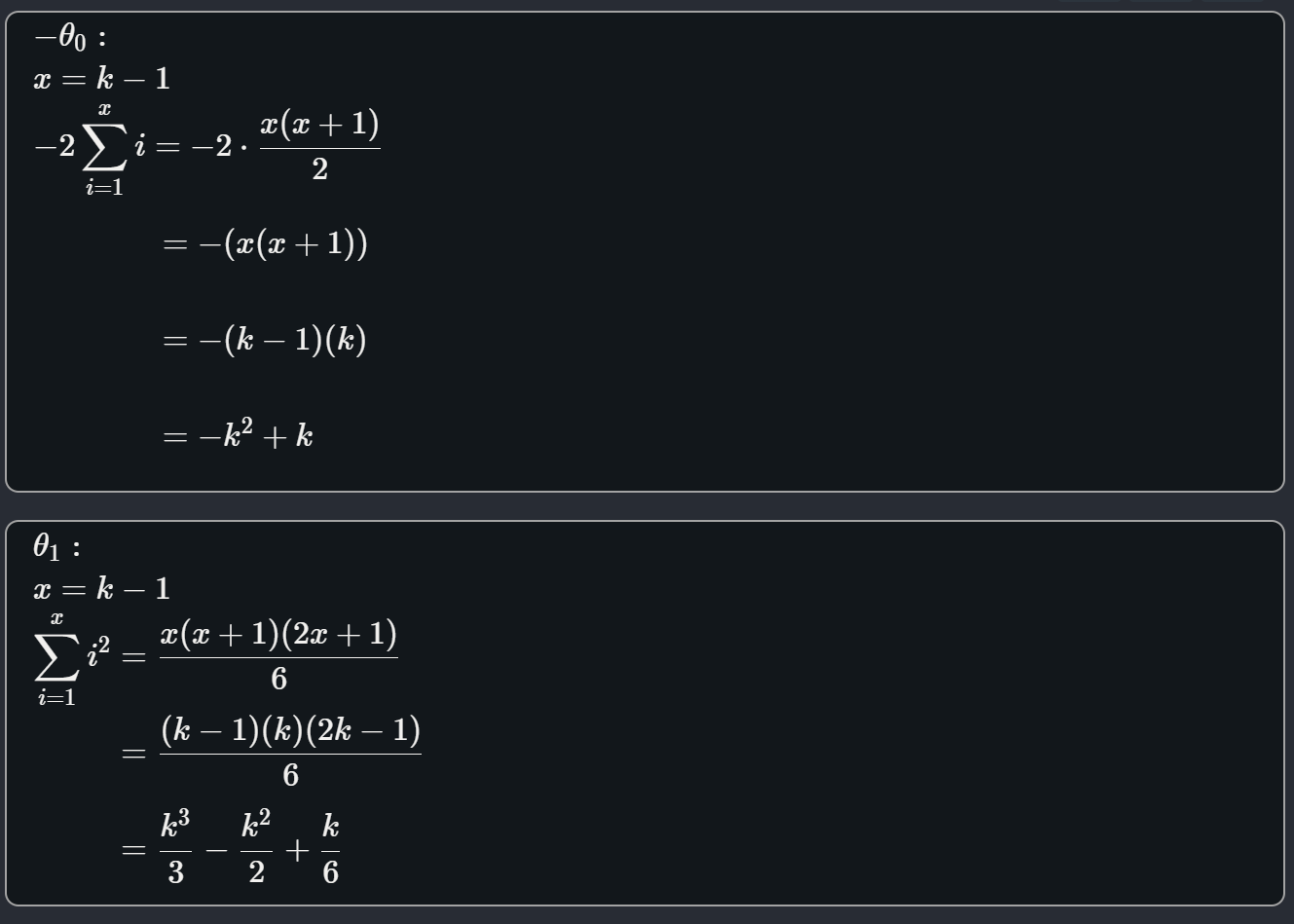
**Tutorial 2**

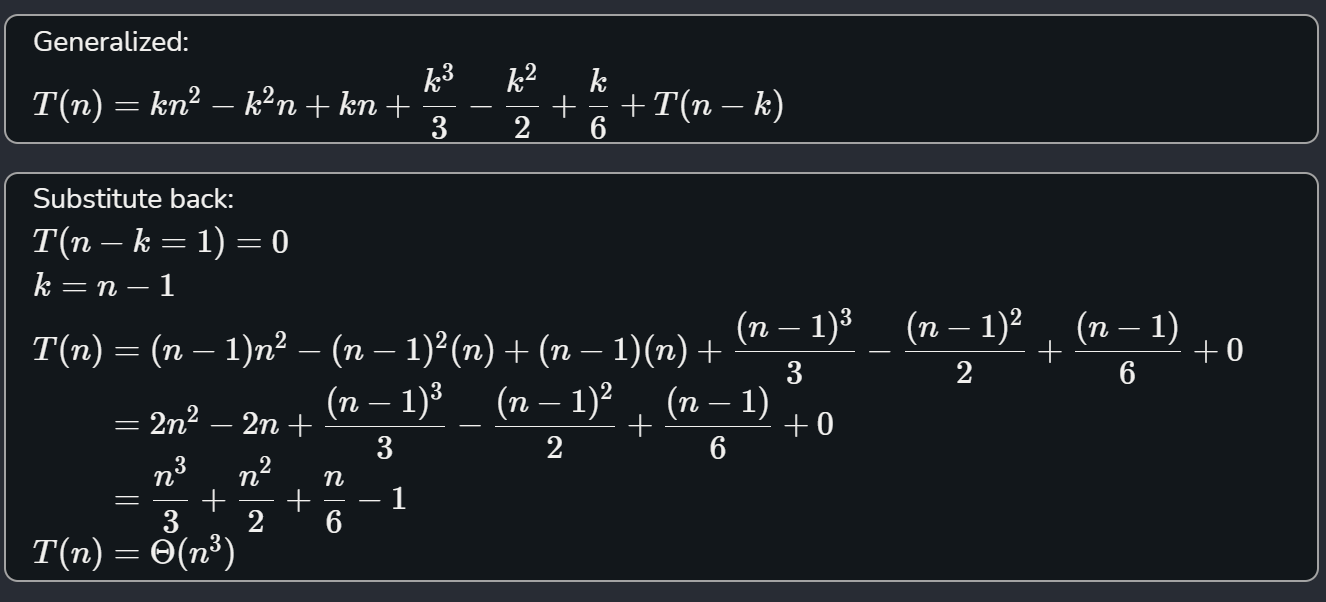
1. a) Using the substitution method, find the time complexity of a recursive program   
 with the following recurrence relation:

T(n) = n2+ T (n-1); n>1

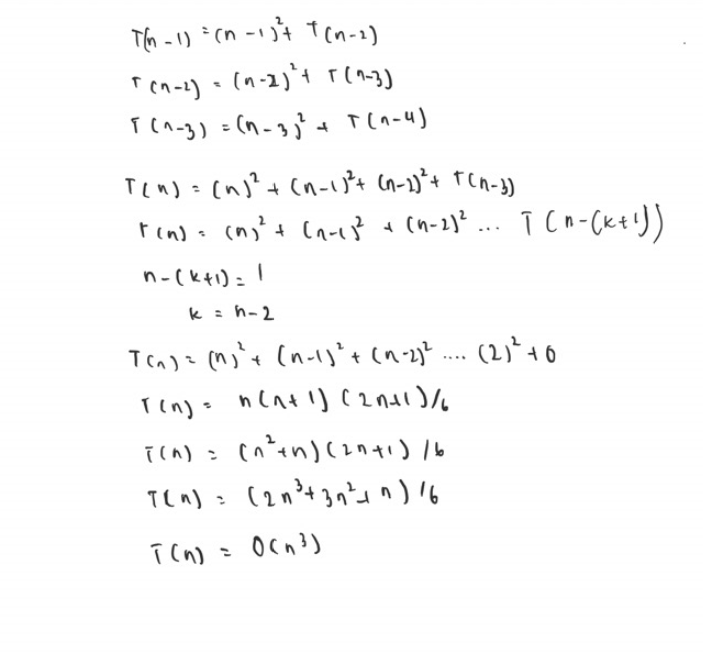
= 0 ; n= 0,1





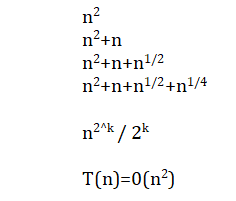


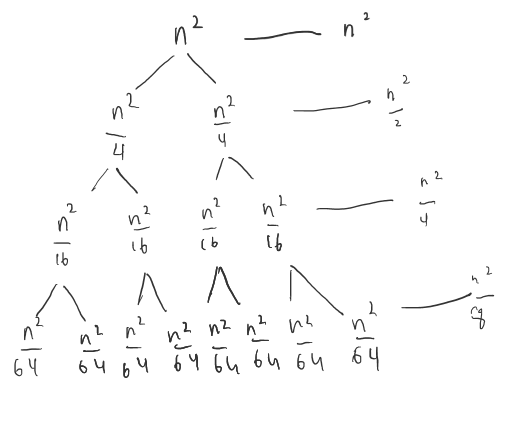
3

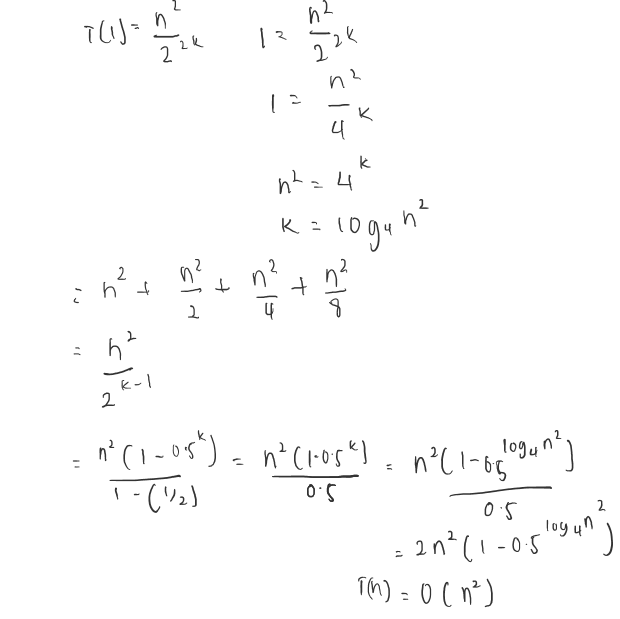


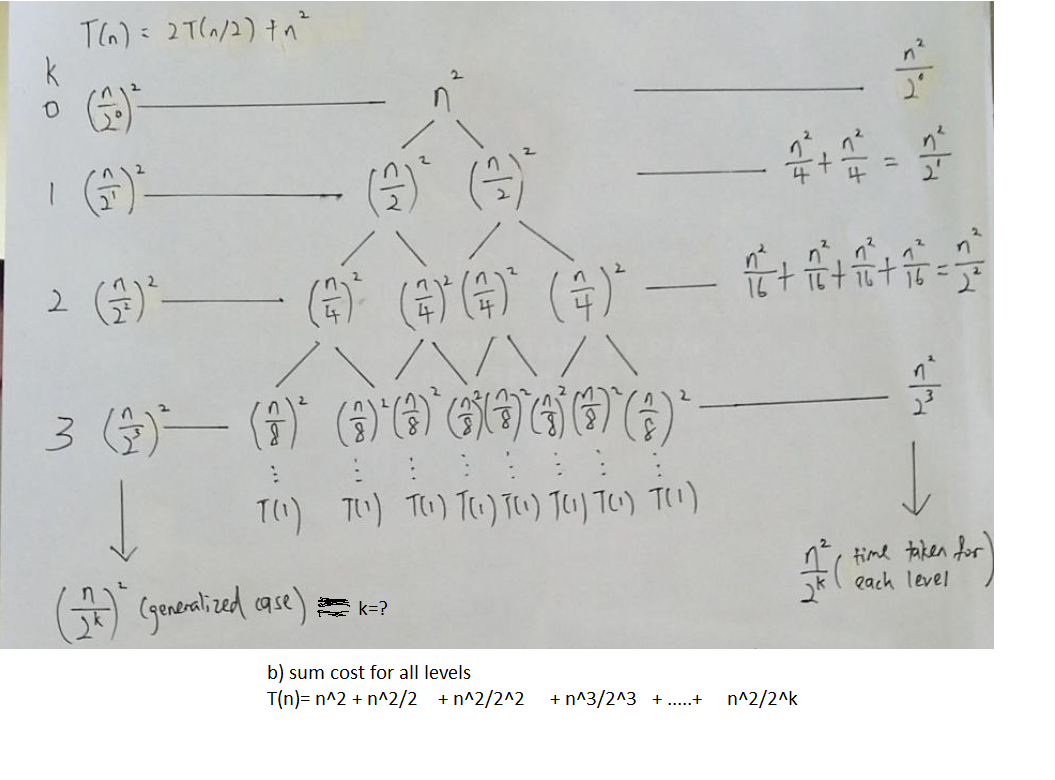
b) Outline the time analysis of the following recursive programs using recursion tree method for

i) T(n) = 2T(n/2) + n2; where n>1



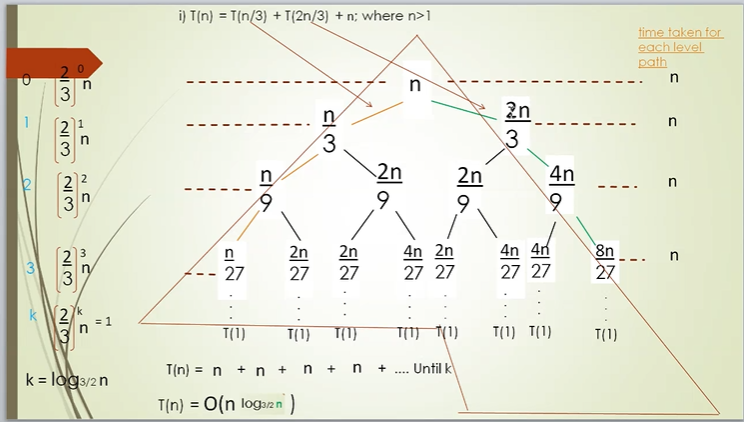






credit to Eng Sheng Hao from T7/T8 :)

ii) T(n) = T(n/3) + T(2n/3) + n; where n>1



2. Using the master methods, solve the following recurrences:

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

a=2 b=4 k=0

2>1

a>b

T(n) =Θ(n^log42)

=Θ(n^1/2)

[HS] Okay

b. T(n) = 2T (n/4) + √n

a=2 b=4 k=0.5

a=b^k

T(n) =Θ(√n logn)

[HS] Okay

c. T(n) = 2T (n/4) + n

a=2, b=4, k=1

T(n)= Θ(n)

[HS] Okay

d. T(n) = 2T(n/4) + n2

a=2, b=4, k=2, p=0

b^k = 16

a<b^k, p>=0

T(n)= Θ(n^2)

[HS] Okay

e. T(n) = 3T (n/2)+ n2

a=3, b=2, k=2

a<b^k

T(n)= Θ(n^2)

[HS] Okay

f. T(n) = 4T (n/2)+ n2

First way

a = 4, b = 2, k = 2, p = 0

a bk

4 = 4

a = bk

Case 2a:

T(n) = Θ(nlog24 log0+1n)

= Θ(n2 logn)

Second way g(n) = nlogba

f(n) = n2 , a=4, b=2, g(n) = nlog24

= n2

f(n) = Θg(n)

Case: 2

T(n) = Θ(nlog24 log n)

= Θ(n2 logn)

g. T(n) = T (n/2) + 2n

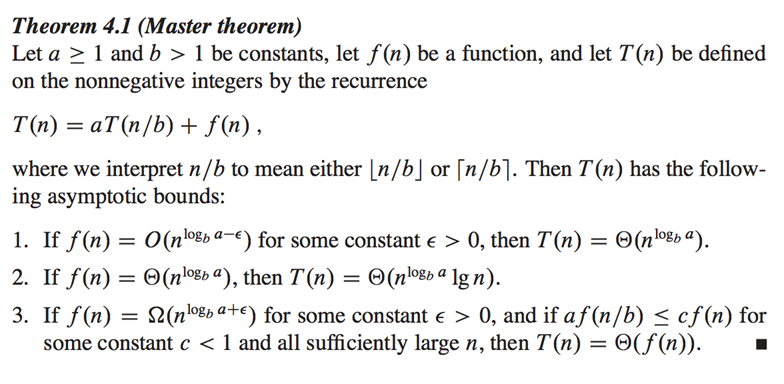
second way =>

a=1 b=2 f(n)=2n

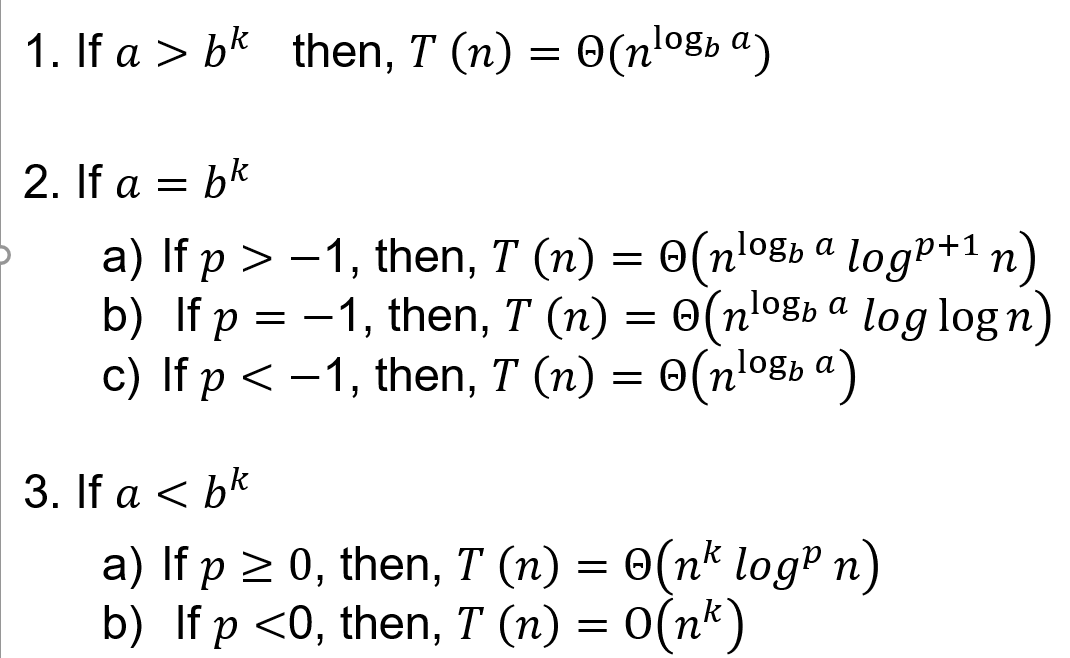
g(n)=nlog22

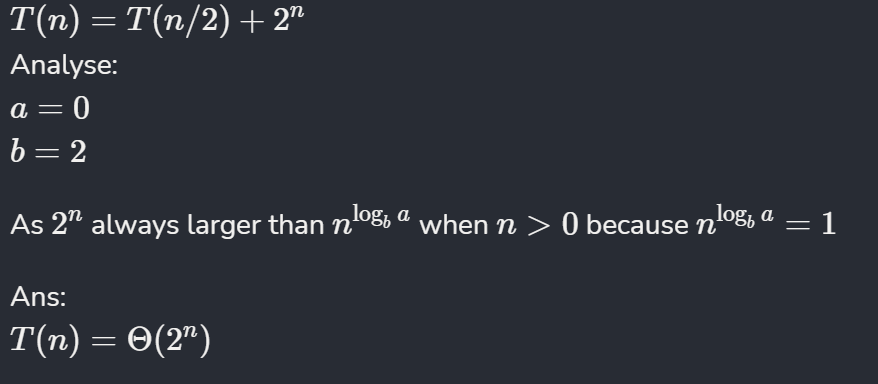
f(n) > g(n) → case 3

T(n)=O(2n)



[HS] someone please answer this :)





h. T(n) = 16T (n/4)+ n

T(n) =Θ(n^2)

i. T (n) = 2T (n/2)+ n log n

a=2 , b=2 ,k=1, p=1

2 = 21

Hence, case 2(a)

T(n) = ɵ(nlogba  log p+1 n)

= ɵ(nlog22  log 2 n)

= ɵ(n log2 n)

T(n) =Θ(n log^2 n)