1 (a)

This factorization will go till ***(n/3i)*** does not become **1**.

If ***n/3i = 1*** then ***n = 3i*** *therefore* ***i = log3(n)***

Hence, there will be *‘i*’ (*log3(n)*) terms in the end.

T(n) = 7n + 91\*(7n/3) + 92\*(7n/3) + ….. + 9i\*(7n/3)

= 7 \* [n + 9\*(n/3) + 92\*(n/32) + ….. + 9i\*(n/3i)]

= 7n \* [1 + 3 + 32 + ….. + 3i]

= 7n \* [1 + 3 + 32 + ….. + 3log3(n)]

= 7n \* [1 + 3 + 32 + ….. + n]

= 7n2 \* [1/3i + 1/3i-1 + 1/3i-2 + ….. + 1]

= 7n2 \* [ 1/(1-(1/3)) ] ]

= 7n2 \* [ 3/(3-1) ]

= 7n2 \* [ 3/2 ]

= (n2)

1 (b)

This factorization will go till ***(n/3i)*** does not become **1**.

If ***n/3i = 1*** then ***n = 3i*** *therefore* ***i = log3(n)***

Hence, there will be *‘i*’ (*log3(n)*) terms in the end.

T(n) = 5n2 + 91\*5(n/3)2 + 92\*5(n/32)2 + ….. + 9i\*5(n/3i)2

= 5 \* [n2 + 9\*(n2/9) + 92\*(n2/92) + ….. + 9i\*(n2/9i)]

= 5n2 \* [1 + 1 + 12 + ….. + 1i] …(i times)

= 5n2 \* [1 + 1 + 12 + ….. + 1log3(n)]

= 5n2 \* [log3(n)]

= (n2 log3(n))

1 (c)

We can see that this factorization will be skewed instead of being symmetric. The (2n/3) factor will take longer time to become 1 as compare to the (n/3) term.

(n/3) term will become 1 when ***n/3i = 1,*** then ***n = 3i*** *therefore* ***i = log3(n)***

From the diagram, it is clear for the first ‘i’ terms will give out **6n**.

Thus, at the end of first ‘i’ terms addition would be 6n\*i = 6n \* log3(n)

For the remaining terms the addition would not go beyond addition of first ‘i’ terms and thus 2 times

value of addition of first ‘i’ terms would upper bound.

Thus,

T(n) = 6n \* log3(n) + Constant

<= 2 \* 6n \* [log3(n)]

<= 12n \* [log3(n)]

<= 12n[log3(n)]

= (nlog3(n))

2 (a)

T(n) = 3T + n2

Will prove this by induction.

For base case :

T(1) = 1 -----<1>

T(2) = 3T(1) + (22) = 3 + 4 = 7 -----<2>

T(3) = 3T(1) + (32) = 3 + 9 = 12 -----<3>

For smaller values of ‘n’ we can say that

***T(1) < T(2) < T(3)***

For larger values :

1 < k < n

T(k) = 3T + k2 ---<4>

T(k + 1) = 3T + (k + 1)2 = 3T + (k2 + 2k + 1) ---<5>

T(k - 1) = 3T + (k - 1)2 = 3T + (k2 - 2k + 1) ---<6>

Now let`s compare equation 4,5 and 6

Since we are taking the floor of the values,

thus <= and same way >= ---<7>

For 4, 5

T(k) ? T(k + 1)

Using equation 7, to compare functions we just need to compare the right side.

k2 ? k2 + 2k + 1

0 < 2k + 1

T(k) < T(k + 1) ---<8>

For 5, 6

T(k) ? T(k - 1)

Using equation 7, to compare functions we just need to compare the right side.

k2 ? k2 - 2k + 1

0 < - 2k + 1

2k > 1

T(k) > T(k - 1) ---<9>

Thus, from 8 and 9 we can say that,

***T(k - 1) < T(k) < T(k + 1)***

As the value of function increases with the increase in value of ‘n’,

we can say that the function is non-decreasing.

2 (b)

T(n) = 3T + n2

This factorization will go till ***(n/3i)*** does not become **1**.

If ***n/2i = 1*** then ***n = 2i*** *therefore* ***i = log2(n)***

Hence, there will be *‘i*’ (*log2(n)*) terms in the end.

T(n) = n2 + (3/4)n2 + (3/4)2n2 + (3/4)3n2 + … + (3/4)in2

T(n) = n2 \* [ 1 + (3/4) + (3/4)2 + (3/4)3 + … + (3/4)i]

T(n) <= n2 \* 2

T(n) <= 2n2

**T(n) = (n2)**

Thus, we need to prove that there exists C1, C2 > 0 and no > 0, such that for all values of n > no,

C2(n2) <= T(n) <= C2(n2)

For base case :

T(1) = 1