**(1) Express the function n3/1000 - 100n2 - 100n + 3 in terms of ‚Ɵ-notation.**

* Let f(n) = n3/1000 - 100n2 - 100n + 3

We consider the highest power of the equation and ignore the constant associated with that term. Thus, f(n) = Ɵ(n3*)*

**(2) Let f(n) and g(n) be asymptotically nonnegative functions. Using the basic definition**

**of ‚ Ɵ -notation, prove that max( f(n) , g(n) ) = Ɵ( f(n) + g(n))**

* We can assume that there exist n1,n2 such that

f(n) >= 0 ; n>n1

g(n) >= 0 ; n>n2

Now, let n0 = max(n1,n2) ; n>n0

f(n)=< max( f(n) , g(n) )

g(n)=< max( f(n) , g(n) )

[ f(n) + g(n) ] =< 2 [ max( f(n) , g(n) ) ]

max( f(n) , g(n) ) =< f(n) + g(n)

Thus we can prove that

0=< (1/2) [ f(n) + g(n) ] =< min( f(n) , g(n) ) =< f(n) + g(n) ; n>n0

This is definition of Ɵ( f(n) + g(n)) where C1= (½) , C2= 1

**(3) Show that for any real constants a and b, where b > 0, (n+a)b = Ɵ(nb)**

* We need to find three constants C1, C2, n0 such that

0=<C1nb =< (n+a)b =< C2nb ; n>n0

As we know that n>|a| we have,

n+a =< n+|a| < 2n

As n >= 2|a|, we have

n+a >= n -|a| > (1/2)n

Thus, when n >= 2|a|, we have

0 =< (1/2)n =< n+a =< 2n

As we know that b is positive constant

0 =< ((1/2)n)b =< (n+a)b =< (2n)b

0 =< (1/2)b nb =< (n+a)b =< 2b nb

So, C1 = (½), C2= 2b n0 = 2|a|

It satisfies the definition

**(4) Let f(n) and g(n) be asymptotically positive functions. Prove or disprove each of**

**the following conjectures.**

* **f(n) + g(n) = Ɵ(min f(n) , g(n))**

Let f(n) = n and g(n) = n2

We know that n2 + n ≠ Ɵ(min (n2, n)) = Ɵ(n)

Thus INCORRECT

* **f(n) + o(f(n)) = Ɵ (f(n))**

According to definition,

0 =< o(f(n)) =< f(n), Hence f(n) =< f(n) + o(f(n)) =< 2(f(n))

Therefore, we can say that f(n) + o(f(n)) = Ɵ (f(n)). CORRECT

**(5) Use a recursion tree to determine a good asymptotic upper bound on the recurrence**

**T(n) = T(n/2) + n2. Use the substitution method to verify your answer.**

* Rate of increase in number of subproblems in each recursion = 1

Rate of decrease in subproblem size = 2

Hence in each level of the tree, there is only one node of cost c(n/2i)2 at depth i=0,1,2,…,lgn.

Hence, total cost of the tree is:

T(n) = (n/2i)2

= cn2 i

= cn2  i

= O(n2)

Now the sum is independent of n, resulting in a constant factor.

We must prove the same using substitution method. Thus

T(n) = T(n/2) + n2

=< d(n/2)2 + n2

= d(n2/4) + n2

= n2 ((d/4)+1)

=< dn2

The last step is true if d >= (d/(4+1)), i.e. d>= (4/3)

**(6) Use a recursion tree to determine a good asymptotic upper bound on the recurrence**

**T(n) = 2T (n – 1) + 1. Use the substitution method to verify your answer.**

* The problem size at depth i is n-i.

Thus total cost at level i would be i=0,1,2,3,4…..n-1 ie 2i

T(n) = i

= (2n-1 – 1)/(2-1)

= 2n-1 – 1

= Ɵ(2n)

We guess T(n) =< c2n +n

T(n) =< (2)( c 2n-1)+ (n-1) +1

= c2n + n

= O(2n)

**(7) Can the master method be applied to the recurrence T(n) = 4T (n/2) + n2 lg n?**

**Why or why not? Give an asymptotic upper bound for this recurrence.**

* Here in this problem we have a=4, b=2, f(n)= *n*2lg*n* ​≠ *O*(*n*2−*ϵ*) ≠ Ω(*n*2−*ϵ*)

So, we cannot apply master method.

Let’s assume T(n)=< cn2lg2n,

T(n) =< 4T (n/2) + n2 lg n

= 4*c*(*n*/2)2lg2(*n*/2) + *n*2lg*n*

*= cn2 lg*(*n*/2) lg(n) – cn2 *lg*(*n*/2) lg(2) + n2lg(n)

= *cn*2lg2*n*−*cn*2lg(*n)* lg(2) - cn2 *lg*(*n*/2) + n2lg(n)

= *cn*2lg2*n + (1-c)n2lg(n) -* cn2 *lg*(*n*/2) (c>1)

=< *cn*2lg2*n -* cn2 *lg*(*n*/2)

=< *cn*2lg2*n*

**(8) Give asymptotic upper and lower bounds for T(n) in each of the following recurrences. Assume that T(n) is constant for n =< 2. Make your bounds as tight as possible and justify your answers.**

* The master method applies to recurrences of the form

T(n) = aT(n/b) + f(n) -----🡪 Compare Following equation with all

where a≥1, b> 1, and f is asymptotically positive.

* **T(n) = 2T(n/2) + n4.**

By master theorem, T(n) = Θ(n4)

* **T(n) = T(7n/10) + n.**

By master theorem, T(n) = Θ(n)

* **T(n) = 16T(n/4) + n2.**

By master theorem, T(n) = Θ(n2lg n)

* **T(n) = 7T(n/3) + n2.**

By master theorem, T(n) = Θ(n2)

* **T(n) = 7T(n/2) + n2.**

By master theorem, T(n) = Θ(nlg 7 )

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

By master theorem, T(n) = Θ(n1/2 lg n)

* **T(n) = T(n-2) + n2.**

Let d=m mod 2

T(n) = 2

= j2 + 4jd + d2

= [( (n)(n+2)(n+1) )/6] + [( n(n+2)d )/2] + [(d2n)/6]

= Θ (n3)