**Question 2:**

1-)

As the master theorem dictates, a >= 1. But in this question, it is 0.5. So it cannot be applied

2-)

3-)

a = 3 b = 3 d = 1

a =

3 = 3

So, it is

= Theta(logn)

= Theta(nlogn)

4-)

a = 6 b = 3

f(n) = Omega()

epsilon > 0

1< -> we got here

a.f() < cf(n) c < 1

5-)

6-)

is not constant. Master theorem cannot be applied

**Question 3:**

**def** chocolate Algorithm(n):  
 **if** n==1:  
 **return** 1  
 **else**:  
 **return** chocolate Algorithm(n-1) + 2 \* n - 1

a-)

This function returns the square of a positive integer n

F(n) = F(n-1) + 2n -1

F(1) = 1

F(2) = 1 + 4 - 1 = 4

F(3) = 4 + 6 – 1 = 9

F(4) = 9 + 8 – 1 = 16

Goes on…

It returns square of input n

We have this

F(n) =

And this

F(n-1) + 2n – 1 =

Which are the same results.

b-)

There is one multiplication in the function.

G(n) = G(n-1) + 1

= (G(n-2) +1) +1

= (G(n-3) +1) + 1\*2

...

…

…

= G(n-i) + 1\*i = G(n-i) + i

G(1) = 0

Let i = n-1 and write G(n-i) + i again

G(n - n +1) + n – 1 = G(1) + n -1 = n – 1

G(n) = n - 1

c-)

There are three addition and substraction in the function

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

= T(n-2) + 3 + 3

= T(n-3) + 3 + 3.2

…

…

…

= T(n-i) + 3.i

Let i = n -1 again like we did above

Then we have T(n-n+1) + 3n – 3

Therefore it is T(1) + 3n -3 and T(1) = 0

T(n) = 3n - 3