Calcular el Zeta de los siguientes códigos, así como establecer cuáles son las dos funciones cota que lo delimitan:

1. x=x+9;

***O(1) = 1***

1. i=1;

while (i<=n) do {

x=x+9;

i=i+5;

}

***1+n+n(1+1)+1 = 2+3n = O(n)***

1. for(i=1;i<=n;i++)

for(j=1;j<=n;j++)

x=x+10;

***1+n+n(1+n+n+1) = 1+n+n+n^2+n^2+n = 1+3n+2n^2 = O(n^2)***

1. int recursiva1(int n){

if(n<=1) return 0;

else return recursiva1(n-1)+recursiva1(n-1);

{

***|-0 si n<=1 -> O(1)***

***T(n) |***

***|-T(n-1)+T(n-1) = 2T(n-1) = 2[2T(n-2)] = 4T(n-2) = …= 2^x\*T(n-x) El último termino es cuando x=n -> 2^n\*T(n-n) -> 2^n\*1 -> 2^n porque T(n-n) tiene un orden de complejidad constante.***

1. int recursiva2(int n){

if(n<=1) return 0;

else return 2\*recursiva2(n /2);

{

***|-0 si n<=1 O(1)***

***T(n) |***

***|-2T(n/2) = 2^xT(n/2^x) -> n/2^x = 1 -> O(log2 (n)) \* O(1) = O(log2 (n))***