

Estructuras de datos y algoritmos: Taller 04

Por: Brahyán Esteban Ríos Soto y Jonathan Smith Julio Díaz.

Punto 1.

```
public class Punto1
{
    public static int maximo(int[] nums, int start){
        if(start == nums.length-1)//C1
        return nums[start]; //T(n)=C1+C2
        else return Math.max(nums[start],maximo(nums,start+1)); //T(n)=C3+T(n-1)
    }
}
```

Tenemos aquí el código separado línea por línea para así encontrar el valor de complejidad de cada una. Ahora, lo que buscamos es la complejidad asintótica esto lo haremos despejando t , usando la herramienta **Wolfram Alpha** se lograra este cometido como se muestra a continuación.



Extended Keyboard

Upload

Examples

Random

Input:

$T(n) = C_3 + T(n - 1)$

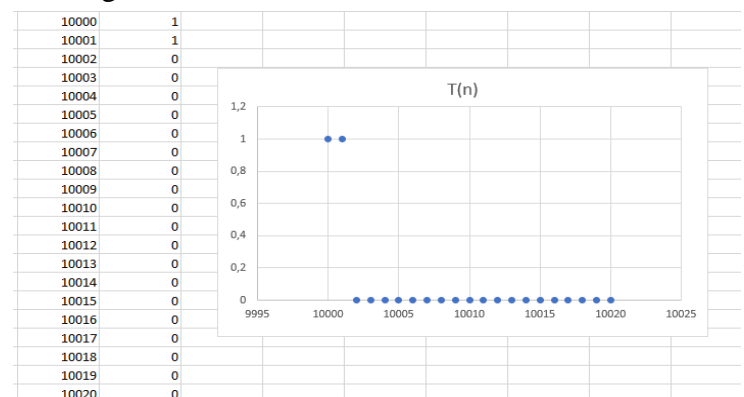
Recurrence equation solution:

$T(n) = C_3 n + c_1$ (where c_1 is an arbitrary parameter)

Podemos observar que arroja que la complejidad se comporta linealmente con una pendiente dependiendo de las operaciones en el caso recursivo y un

intercepto en el caso base. Comprobémoslo con una gráfica.

La grafica lo que nos muestra es que el tiempo empleado no crece consistentemente y tiene unos pequeños peldaños que deben venir de PC mas que de un real aumento en el uso de recursos.



Punto 2.

```
public class Punto2
{
    public static boolean sumaGrupo(int start, int[] nums, int target){
        if(start == nums.length)//C1
            return target == 0;//C2
        else
            return sumaGrupo(start+1, nums, target) ||
                sumaGrupo(start+1, nums, target - nums[start]); //t(n)=C3+2t(n-1)
    }
}
```

Haremos todo análogamente al punto anterior



$t(n) = C_3 + 2t(n-1)$

Extended Keyboard Upload Examples Random

Input:

$t(n) = C_3 + 2t(n-1)$

Recurrence equation solution:

$t(n) = C_3(2^n - 1) + c_1 2^{n-1}$ (where c_1 is an arbitrary parameter)

Lo que nos arroja es que la complejidad es exponencial, esto lo comprobaremos recopilando el tiempo de diferentes ejecuciones y graficando.

Podemos observar en la gráfica un comportamiento exponencial se busco no usar datos muy bajos ya que estos tienden a 0 o 1 y esto no es lo que buscamos, usamos los datos mas altos posibles en un equilibrio entre precisión y tiempo.

