

ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

LUIS ALEJANDRO ARDILA GONZALEZ

DDYA

SEMANA 2

EJERCICIOS

1.

1. $\log(\log(n))$

2. $\log(n)$

3. $\ln(n)$

4. $\log(n)^{**2}$

5. $n^{**(\frac{1}{2})}$

6. n

7. $n\log(n)$

8. $n^{**(1+E)}$ con $0 < E < 1$

9. n^{**2}

10. n^{**3}

11. n^{**5}

12. $2^{**(n-1)}$

13. $n - n^{**3} + 7n^{**5}$

14. 2^{**n}

15. e^{**n}

16. $n! 2.$

2.

2. Para las siguientes funciones, determinar el resultado como una función de n y representar el peor caso de ejecución con notación Big Oh:

<code>function mystery(n)</code>	<code>function pesky(n)</code>	<code>function prestiferous(n)</code>
<pre>r := 0 for i := 1 to n - 1 do for j := i + 1 to n do for k := 1 to j do r := r + 1 return(r)</pre>	<pre>r := 0 for i := 1 to n do for j := 1 to i do for k := j to i + j do r := r + 1 return(r)</pre>	<pre>r := 0 for i := 1 to n do for j := 1 to i do for k := j to i + j do for l := 1 to i + j - k do r := r + 1 return(r)</pre>



$$\sum_{x=1}^n x = \frac{1}{2} n(n+1)$$

$$\sum_{j=i+1}^n j = \sum_{j=1}^n j - \sum_{j=1}^i j$$

$$\sum_{x=1}^n x^2 = \frac{1}{6} n(n+1)(2n+1)$$

-Mistery(n): $T(n)=O(n^{**3})$

-Pesky(n): $T(n)=O(n^{**3})$

-Prestiferous(n): $T(n)=O(n^{**4})$

La elevación depende de la cantidad de anidaciones

3.

```
def insertion_sort(lista: list) -> list:
```

```
    for i in range(1, len(lista)):
```

```
        key = lista[i]
```

```
        j = i - 1
```

```
        while j >= 0 and lista[j] < key:
```

```
            lista[j + 1] = lista[j]
```

```
            j -= 1
```

```
        lista[j + 1] = key
```

```
    return lista
```

```
def main():
```

```
    array = [1, 2, 3, 4, 5]
```

```
    sorter_array = insertion_sort(array)
```

```
    print("La lista ordenada de forma descendente: ", sorter_array)
```

```
main()
```