



Módulo II - Análisis de Algoritmos

Programación III
Ingeniería en Computación – UNLP

12 de junio de 2023

Nombre y apellido	Legajo	Corrigió
EJERCICIO 1:	EJERCICIO 2:	NOTA:

EJERCICIO 1: Puntaje 5 puntos

a. Dado el siguiente algoritmo indique el $T(n)$ correspondiente

```
public void calculo(int n){  
    int x=1;  
    for ( int i=1; i <= n; i++)  
        for ( int j=i2; j >= 1; j--)  
            x=x*2;  
    for (int k=0; k <= n; k++)  
        x=x-2;  
    return x;  
}
```

b. Indique el Orden del $T(n)$ calculado, y usando la definición de Big-OH, demuestre que dicho Orden es correcto.



EJERCICIO 2: Puntaje 5 puntos

1) Considere la siguiente recurrencia:

$$T(n) = 1 \quad \text{si } n=1$$

$$T(n) = 2 \cdot T(n-1) + \log(n) + (n-1) \quad \text{si } n \geq 2$$

¿Cómo se reemplaza $T(n-1)$ considerando $n-1 > 1$

- (a) $2 \cdot T(n-2) + \log(n) + (n-1)$
- (b) $2 \cdot T(n-1) + \log(n-1) + (n-2)$
- (c) $2 \cdot T(n-2) + \log(n-1) + (n-1)$
- (d) $2 \cdot T(n-2) + \log(n-1) + (n-2)$
- (e) $2 \cdot T(n-2) + \log(n-2) + (n-1)$
- (f) Ninguna de las opciones anteriores

2) Se tiene un algoritmo que tiene un tiempo de ejecución de $O(n \log_{10} n)$ y para ejecutar un problema de tamaño 100 tarda 1 hora. ¿Cuánto tardará si se duplica el tamaño de la entrada?

- (a) 2 hs.
- (b) 4 hs.
- (c) 1,30 hs.
- (d) 2,30 hs.
- (e) Ninguna de las anteriores

3) El orden de ejecución para **Eliminar** un elemento en un arreglo ordenado, manteniéndolo ordenado, es:

- (a) $O(n^2)$
- (b) $O(\log(n))$
- (c) $O(n)$
- (d) $O(n \cdot \log(n))$
- (e) $O(1)$

4) Considere la siguiente expresión:

$$(3^n + 2)(3^n + 5)(n + 3) \quad \text{¿Cuál es el } O(n)?$$

- (a) $O(9^n)$
- (b) $O(n \cdot 9^n)$
- (c) $O(n \cdot 3^n)$
- (d) $O(n^2 \cdot 3^n)$
- (e) Ninguna de las opciones

5) Dado el siguiente algoritmo

```
void ejercicio (int n) {  
    if (n >= 4) {  
        n = n/4;  
        4 * ejercicio(n);  
        2 * ejercicio(n);  
    }  
}
```

Indique el $T(n)$ para $n \geq 4$

- (a) $T(n) = c + 4 \cdot T(n) + 2 \cdot T(n)$
- (b) $T(n) = c + 4 \cdot T(n/4) + 2 \cdot T(n/4)$
- (c) $T(n) = c + T(n/4) + T(n/4)$
- (d) $T(n) = c + T(n) + T(n)$
- (e) $T(n) = c + 8 \cdot T(n/4)$

Tema 2

Parte Práctica

Apellido	Nombre	Legajo

Ejercicio 1	Corrigió

EJERCICIO 1: Puntaje 5 puntos

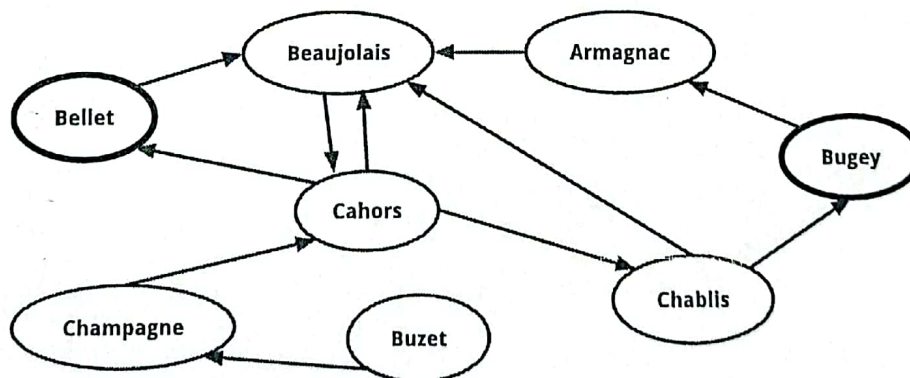
Se cuenta con un mapa de pueblos emblemáticos de Francia y se quiere conocer todos los caminos desde un pueblo origen hasta un pueblo destino, teniendo en cuenta que no nos interesa pasar por algunos pueblos específicos pasados en una lista como parámetro.

Tenga en cuenta que:

- Debe devolver todos los caminos posibles, desde un pueblo origen hasta un pueblo destino, evitando los pueblos pasados por parámetro.
- Debe completar en la firma del método los tipos de datos indicados con signo de interrogación.
- Debe verificar la existencia del pueblo origen y del pueblo destino.
- **No se puede pasar 2 veces por el mismo lugar** al formar cada recorrido o camino.
- En caso de no existir un recorrido posible, debe devolver la **lista vacía**.
- **Debe elegir alguno de los recorridos vistos en clase: DFS o BFS**

Implemente la clase **Parcial**, y el método:

??? resolver(Grafo<???> ciudades, String origen, String destino, Lista<???> evitarPasandoPor)



En este ejemplo, si el pueblo **origen** es **Buzet**, el **destino** es **Beaujolais** y los pueblos a evitar son **Bellet y Bugey**, los recorridos resultantes serían:

Buzet > Champagne > Cahors > Beaujolais

Buzet > Champagne > Cahors > Chablis > Beaujolais

Si bien Buzet > Champagne > Cahors > Bellet > Beaujolais tiene el origen y destino solicitado, el camino está pasando por Bellet que está en la lista de pueblos a evitar.

Tema 2

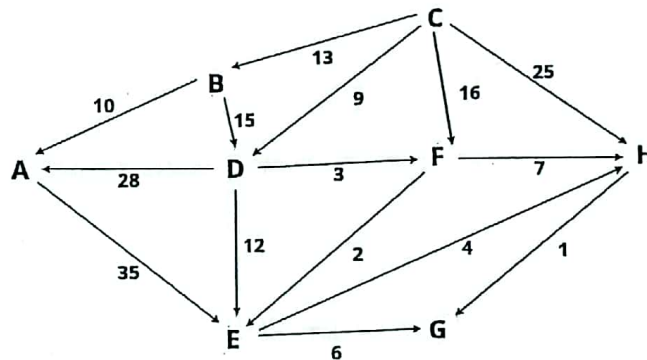
Parte Teórica

Apellido	Nombre	Legajo	Corrigió

Ejercicio 2	Ejercicio 3

EJERCICIO 2: Puntaje 3 puntos

(a) Se comenzó a ejecutar el algoritmo de Dijkstra sobre el siguiente dígrafo pesado, a partir del vértice 'C', continúe con la ejecución hasta su finalización.

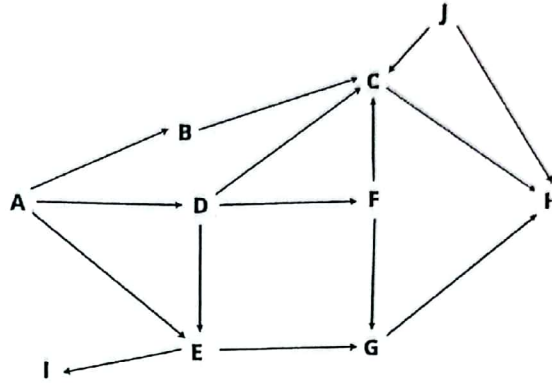


Muestre todos los pasos intermedios, indicando el orden en que se van procesando los vértices.

Orden en que toma el vértice	Vértices v	Costo (A,v)	Previo	Visitado
	A	∞		0
	B	∞ 13	C	0
1º	C	0	---	0 1
	D	∞ 9	C	0
	E	∞		0
	F	∞ 16	C	0
	G	∞		0
	H	∞ 25	C	0

EJERCICIO 3: Puntaje 2 puntos

Obtener la ordenación topológica para el siguiente grafo dirigido acíclico, utilice la estrategia que trabaja con los grados de entrada de los vértices y utiliza una Cola. Muestre la ejecución del algoritmo indicando en cada paso cómo van evolucionando los grados de entrada de los vértices y la estructura de Cola. Nota: considere que los vértices y las listas de adyacentes están ordenadas alfabéticamente.



Área reservada para mostrar la ejecución del algoritmo de ordenación topológica, incluyendo los grados de entrada de los vértices y la estructura de la Cola en cada paso.