

[EJERCICIO 1](#)

[EJERCICIO 2](#)

[EJERCICIO 3](#)

[EJERCICIO 4](#)

[EJERCICIO 5](#)

Version 1

Explique de qué se trata el **teorema maestro** y cómo se utiliza.

Ejemplifique cada caso que resuelve con una ecuación de recurrencia.

Version 1

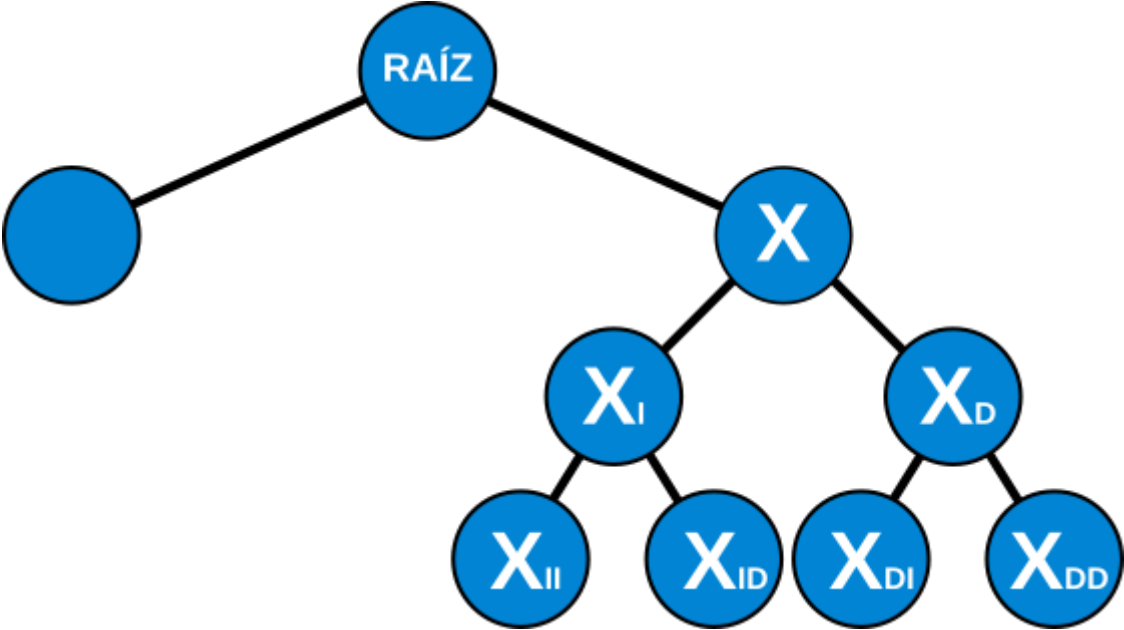
1. Dada la siguiente definición de ABB recursivo:

```
typedef struct nodo{
    void* dato;
    struct nodo* izquierda;
    struct nodo* derecha;
}nodo_t;
```

Implemente la siguiente función:

```
nodo_t* rotar_derecha(nodo_t* nodo_a_rotar);
```

2. Complete el siguiente código con una llamada a la función implementada para rotar el nodo X.



```
int main(){
    nodo_t* arbol = NULL;
    arbol = arbol_crear(/*....*/);
    arbol = arbol_insertar(/*....*/);

    //... mas cosas ...//

    //Acá quiero rotar el nodo X antes de terminar

    <su código aquí>

    arbol_destruir(arbol);
    return 1;
}
```

No es necesario copiar todo el código, solamente escriba el fragmento de código que se pide.

Version 2

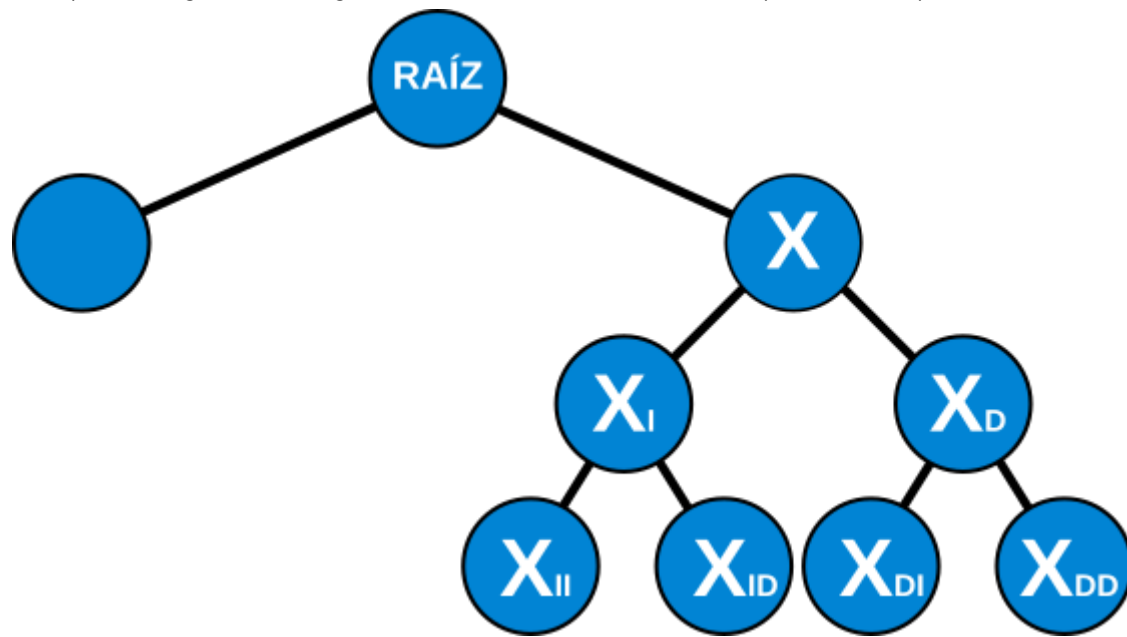
1. Dada la siguiente definición de ABB recursivo:

```
typedef struct nodo{
    void* dato;
    struct nodo* izquierda;
    struct nodo* derecha;
}nodo_t;
```

Implemente la siguiente función:

```
nodo_t* rotar_izquierda(nodo_t* nodo_a_rotar);
```

2. Complete el siguiente código con una llamada a la función implementada para rotar el nodo X.



```
int main(){
    nodo_t* arbol = NULL;
    arbol = arbol_crear(/*....*/);
    arbol = arbol_insertar(/*....*/);

    //... mas cosas ...//

    //Acá quiero rotar el nodo X antes de terminar

    <su código aquí>

    arbol_destruir(arbol);
    return 1;
}
```

No es necesario copiar todo el código, solamente escriba el fragmento de código que se pide.

Version 1

1. Explique si la siguiente estructura puede o no representar algún tipo de **tabla de hash**, y justifique.

6				4	5
---	--	--	--	---	---

En caso negativo, primero explique (y defina) qué es necesario para que pueda ser una **tabla de hash**.

Caracterize la tabla, explique sus propiedades y cómo funciona.

2. Inserte en la tabla los valores 1, 2, 4 y 5, mostrando cómo queda la tabla luego de cada operación.

Version 2

1. Explique si la siguiente estructura puede o no representar algún tipo de **tabla de hash**, y justifique.

	7		3		5
--	---	--	---	--	---

En caso negativo, primero explique (y defina) qué es necesario para que pueda ser una **tabla de hash**.

Caracterize la tabla, explique sus propiedades y cómo funciona.

2. Inserte en la tabla los valores 1, 2, 4 y 5, mostrando cómo queda la tabla luego de cada operación.

Version 3

1. Explique si la siguiente estructura puede o no representar algún tipo de **tabla de hash**, y justifique.

6		2			5
---	--	---	--	--	---

En caso negativo, primero explique (y defina) qué es necesario para que pueda ser una **tabla de hash**.

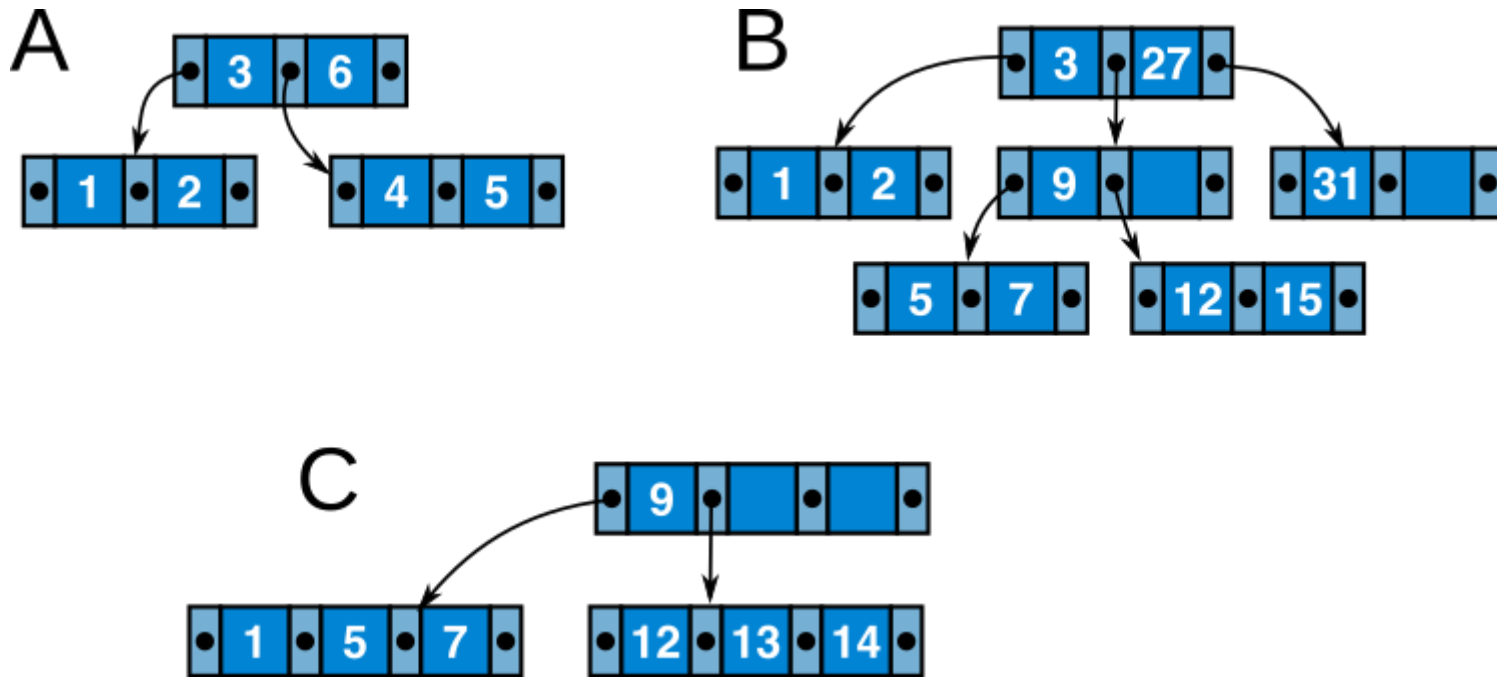
Caracterize la tabla, explique sus propiedades y cómo funciona.

2. Inserte en la tabla los valores 1, 2, 4 y 5, mostrando cómo queda la tabla luego de cada operación.

[EJERCICIO 1](#)[EJERCICIO 2](#)[EJERCICIO 3](#)[EJERCICIO 4](#)[EJERCICIO 5](#)

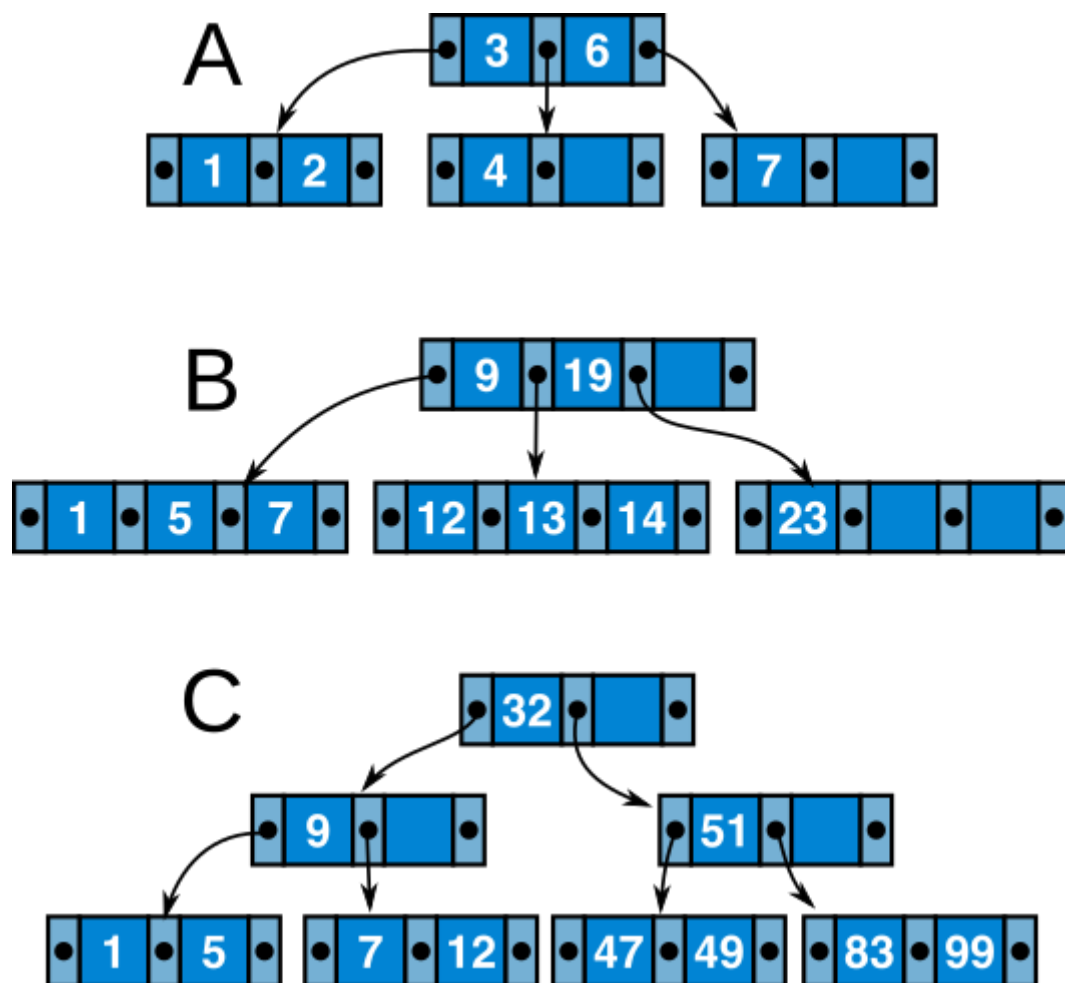
Version 1

Indique si los siguientes son árboles B válidos o no. Justifique.



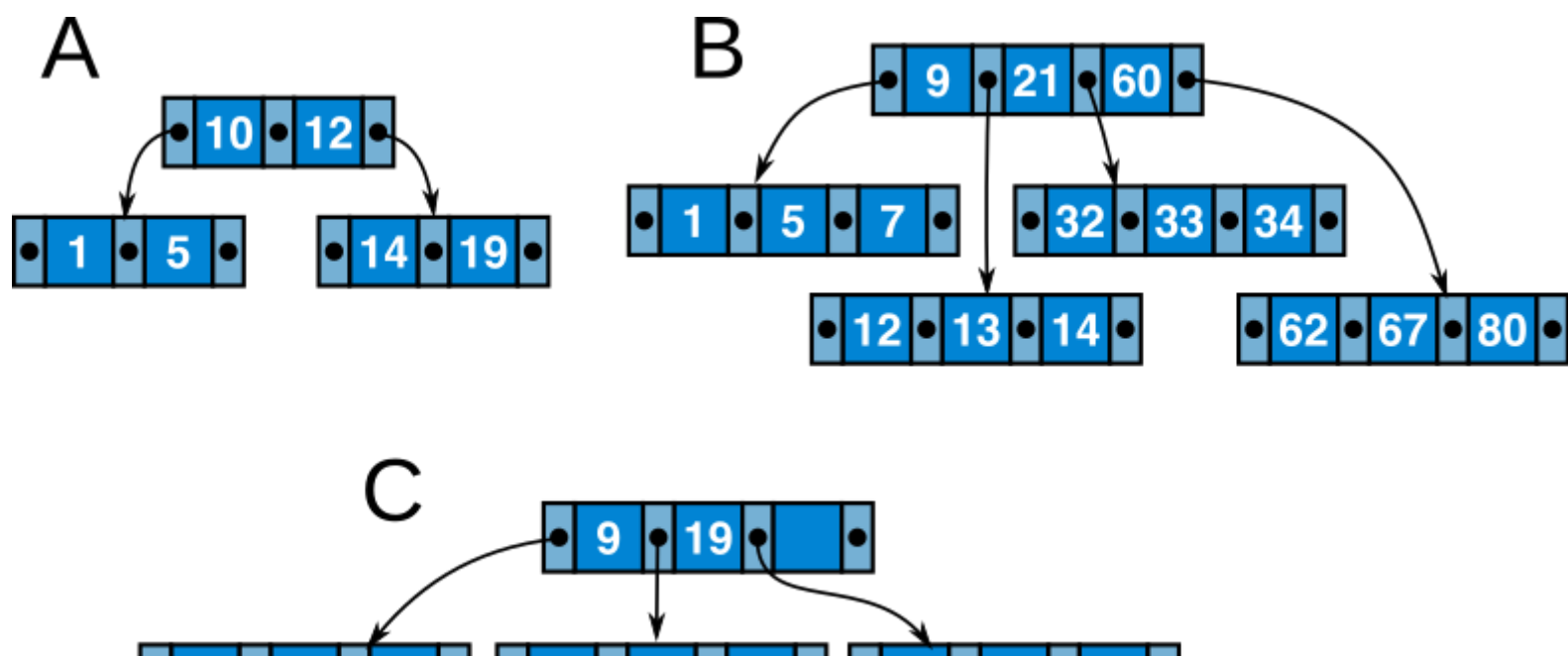
Version 2

Indique si los siguientes son árboles B válidos o no. Justifique.



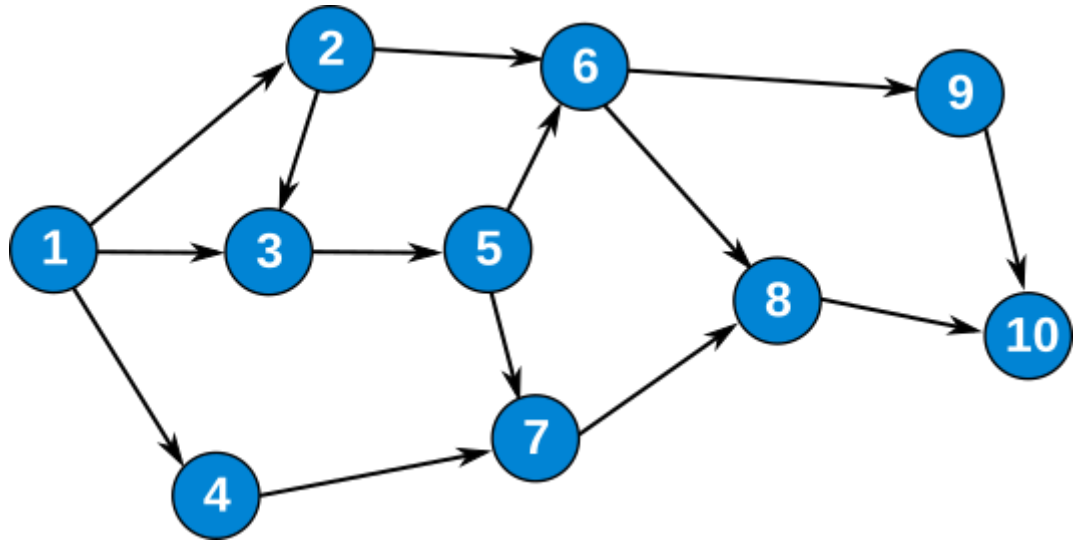
Version 3

Indique si los siguientes son árboles B válidos o no. Justifique.



Version 1

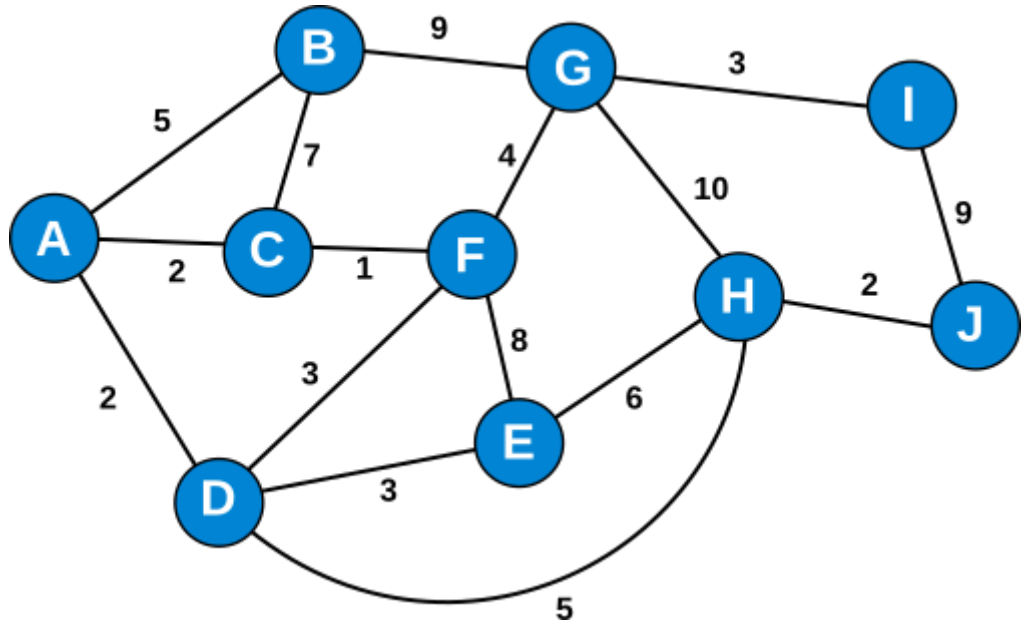
1. Exprese, si es posible, el siguiente grafo como **matriz de adyacencias** y como **matriz de incidencias**. Explique brevemente.



2. Explique como funciona el algoritmo de **Floyd-Warshall** y muestre (en código) cómo podría implementarse (sólo el algoritmo, **NO** el tda grafo completo por favor).

Version 2

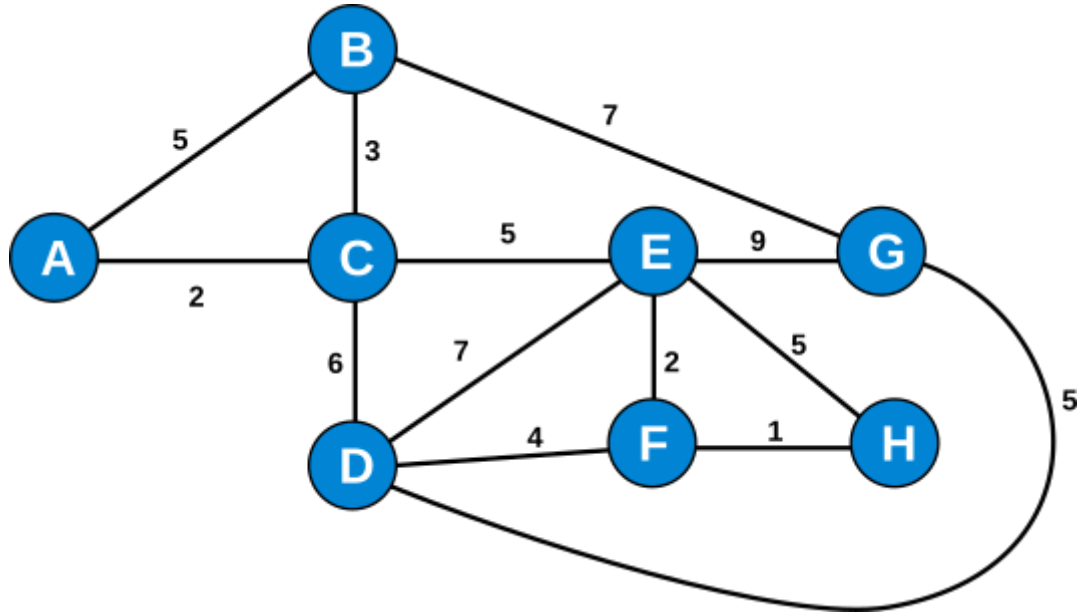
1. Exprese, si es posible, el siguiente grafo como **matriz de adyacencias** y como **matriz de incidencias**. Explique brevemente.



2. Explique como funciona el algoritmo de **Dijkstra** y aplíquelo desde el vértice **A** hasta el **H**.

Version 3

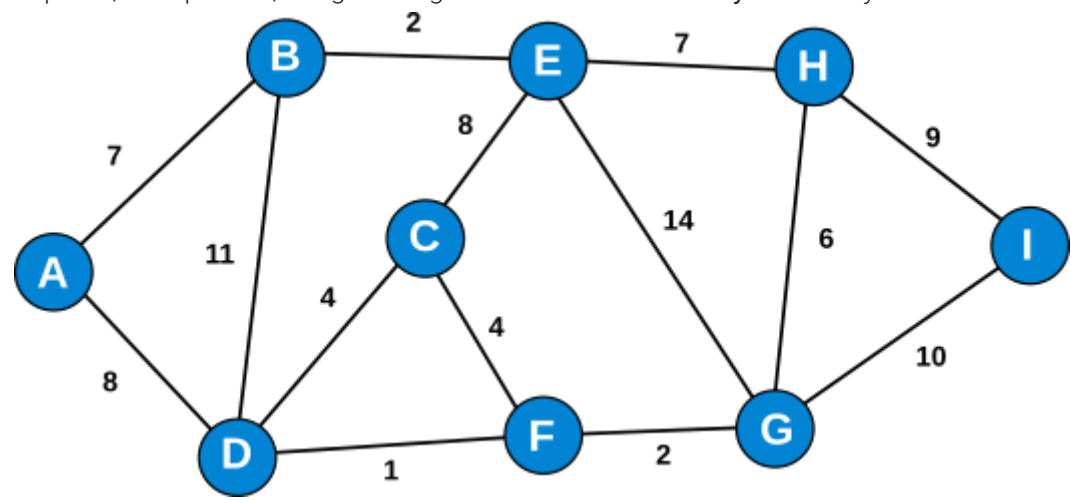
1. Exprese, si es posible, el siguiente grafo como **matriz de adyacencias** y como **matriz de incidencias**. Explique brevemente.



2. Explique como funciona el algoritmo de ***Prim ***, cómo se compara con el de **Kruskal** y aplíquelo empezando por el vértice **C**.

Version 4

1. Exprese, si es posible, el siguiente grafo como **matriz de adyacencias** y como **matriz de incidencias**. Explique brevemente.



2. Explique como funciona el algoritmo de **Kruskal**, cómo se compara con el de **Prim** y aplíquelo.
