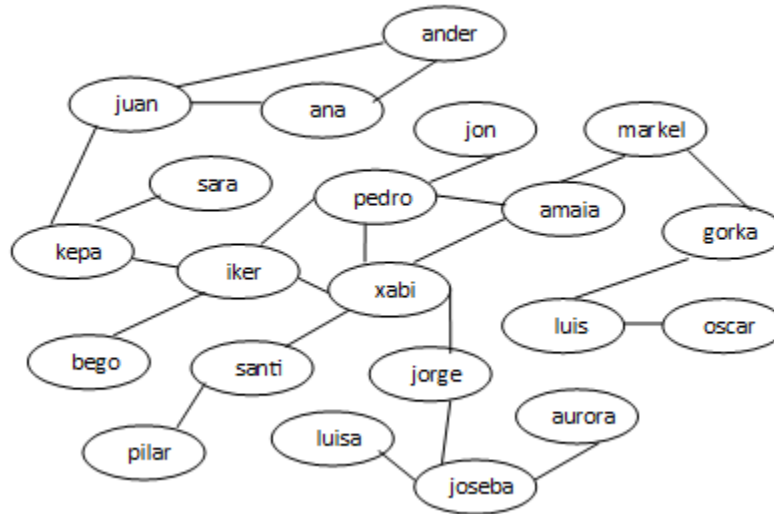


Laboratorio 9: Grafos

Ejercicio 1

Tenemos el siguiente grafo no dirigido, que representa relaciones de parentesco entre personas:



Cuando una persona fallece, se reparte su herencia a partes iguales entre todos los parientes que estén a una distancia máxima *distMax* de la persona fallecida.

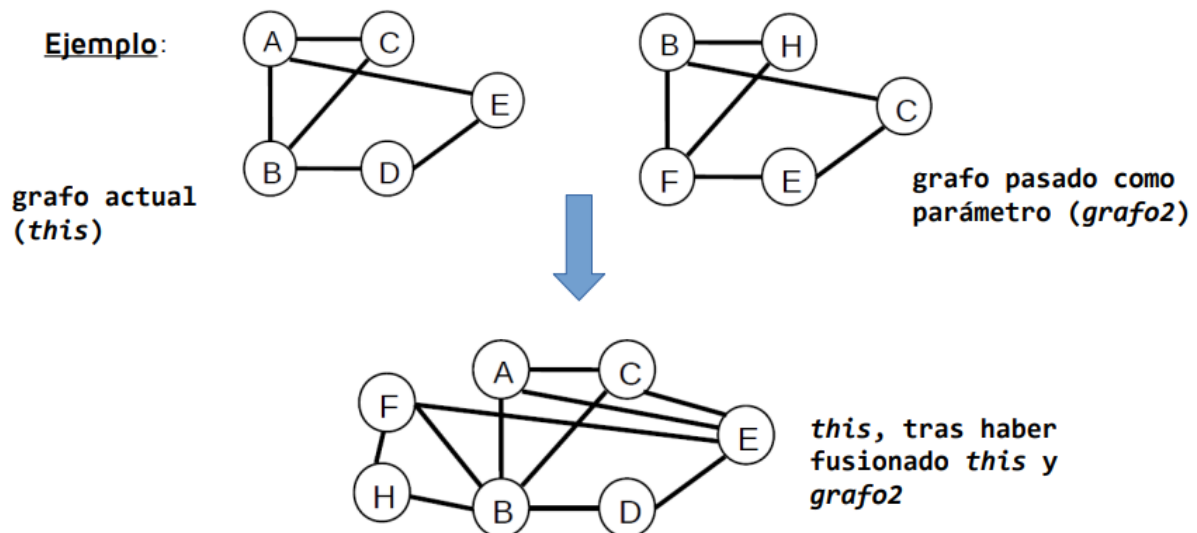
Por ejemplo, si Pedro fallece y queremos repartir 100000€ de herencia entre los parientes a distancia máxima 2, los beneficiarios serían Jon, Amaia, Xabi, Iker (por encontrarse a distancia 1) y Markel, Jorge, Santi, Kepa y Bego (por encontrarse a distancia 2). Cada uno de ellos recibiría $100000/9 = 11111\text{€}$.

Se pide implementar el método *repartirHerencia* de la clase *Herencia*, que dado el nombre de la persona fallecida, el valor de su herencia y una distancia, devuelve la cantidad que recibirá cada uno de los beneficiarios (será igual para todos).

Ejercicio 2.

La clase *GrafoNoDirigidoMap* sirve para representar grafos no dirigidos, no ponderados, implementados mediante un *HashMap* en el que para cada nodo se guarda un *LinkedList* con sus nodos adyacentes.

Se pide implementar el método *fusionar* de la clase *GrafoNoDirigidoMap*. Este método recibe como parámetro un segundo grafo del mismo tipo. El método *fusionar* debe modificar el grafo actual de manera que al terminar contenga la fusión entre el grafo actual y el grafo pasado como parámetro. Debes tener en cuenta que en el grafo resultante no puede haber nodos y aristas repetidos.



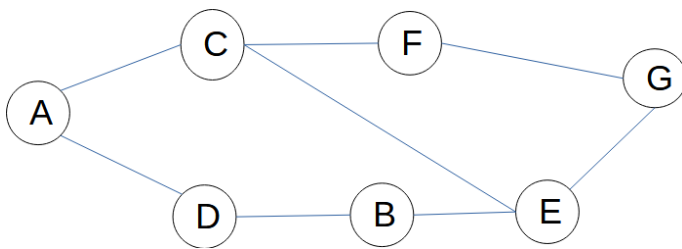
Ejercicio 3

La clase *GrafoNoDirigidoListas* sirve para representar grafos no dirigidos, no ponderados, implementados mediante listas de adyacencias.

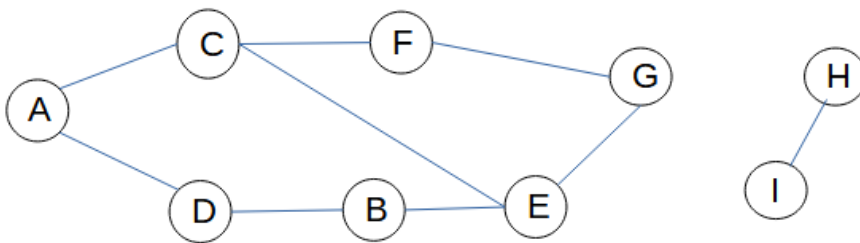
Se pide implementar el método *esConexoAnchura*, que, utilizando un recorrido en anchura devuelve un booleano que indica si el grafo es conexo o no.

Ejemplo

El siguiente grafo sí es conexo:



En cambio, el siguiente grafo no es conexo:



Salida esperada

Ejercicio 1:

Se reparte la herencia de pedro, valorada en 100000€, entre sus parientes a distancia máxima 2:

Cada uno de ellos recibirá 11111€

Se reparte la herencia de kepa, valorada en 300000€, entre sus parientes a distancia máxima 4:

Cada uno de ellos recibirá 20000€

Se reparte la herencia de amaia, valorada en 20000€, entre sus parientes a distancia máxima 8:

Cada uno de ellos recibirá 1000€

Ejercicio 2: (el orden no es relevante)

Grafo 'this' antes de fusionar:

A: B C E

B: A C D

C: A B

D: B E

E: A D

Grafo2 antes de fusionar:

B: H F C

C: B E

E: F C

F: B H E

H: B F

Grafo 'this' después de fusionarlo con grafo2:

A: B C E

B: A C D H F

C: A B E

D: B E

E: A D F C

F: B H E

H: B F

Ejercicio 3:

Grafo:

A: C D

B: D E

C: A E F

D: A B

E: B C G

F: C G

G: E F

Es conexo?: true

Tras añadir los nodos H e I:

Grafo:

A: C D

B: D E

C: A E F

D: A B

E: B C G

F: C G

G: E F

H: I

I: H

Es conexo?: false