ENTREGA E7 (0,5 puntos)

Antes de empezar...

- Descargar el archivo **E7.zip** e importarlo a Eclipse.
- Renombrar el proyecto (tecla F2). Por ejemplo, para el grupo formado por Ana Pérez, Jon Azkue y Miren Landa el nuevo nombre sería: **E7_APerezJAzkueMLanda**.
- En el fichero **componentesGrupo**, escribir los nombres de los componentes del grupo.

Además, tened en cuenta que:

- Se valorará la eficiencia de las soluciones, es decir, además de que sean correctas, deben ser eficientes.
- Aparte de los casos de prueba que se os entregan, se espera que incorporéis casos de prueba adicionales y significativos.
- Para entregarlo, debéis exportar el proyecto y subirlo a eGela.

Fecha límite de entrega: 29/12/2023 a las 23:59

—

Ayuda para importar:

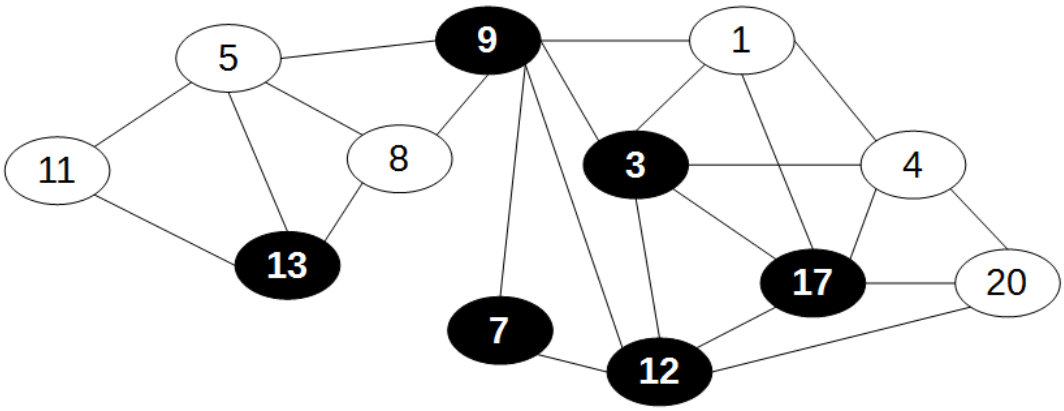
File -> Import... -> General -> Existing Projects into Workspace -> Select archive file (el .zip descargado) -> Finish

Ayuda para exportar:

Pinchar en el proyecto. File -> Export... -> General -> Archive File -> (seleccionar las carpetas/archivos a exportar) -> To archive file (escribir una ruta y nombre para el nuevo archivo .zip) -> Finish

Ejercicio 1

El grafo siguiente representa un juego:



El objetivo del juego es llegar desde una casilla inicial a otra final. Para ello se debe respetar la regla fundamental del juego:

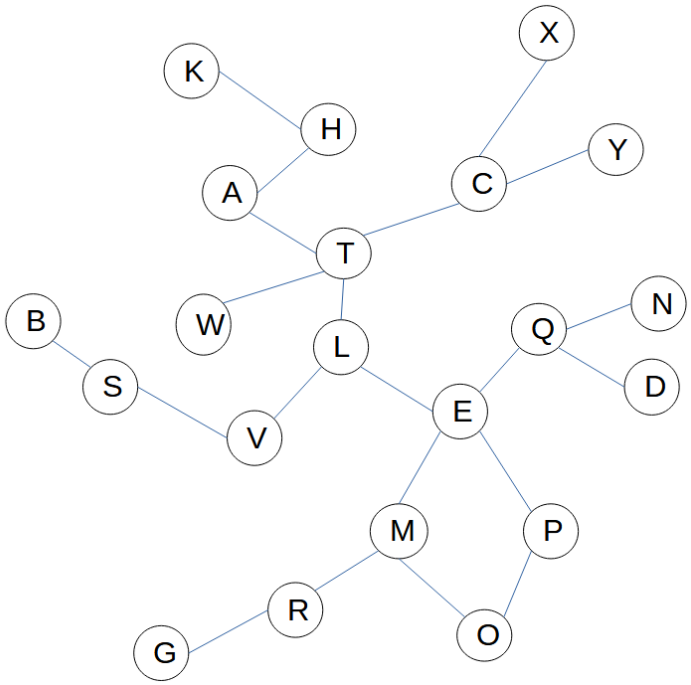
- Desde una casilla de un color solo se puede pasar a casillas de otro color (es decir, solo se puede pasar de blanco a negro o de negro a blanco).

Queremos hacer un algoritmo que, dadas dos casillas del grafo nos devuelva una lista con los elementos del camino más corto que, respetando la regla del juego, conecte las dos casillas.

Por ejemplo, si las casillas inicial y final fueran respectivamente la 11 y la 20, el algoritmo devolvería la lista [11, 13, 5, 9, 1, 17, 20] (o cualquier otro camino de la misma longitud).

Ejercicio 2

Sea un grafo no dirigido cuyos nodos representan ciudades. Las ciudades que son adyacentes entre sí están unidas por una arista. Suponemos que no hay nombres de ciudad repetidos.



Cuando se produce un terremoto de intensidad N en una ciudad, el terremoto se propaga a con una intensidad de N/2 a las ciudades que están a distancia 1 de la ciudad epicentro del terremoto, con una intensidad de N/4 a las que están a distancia 2, N/8 a las que están a distancia 3, y así sucesivamente.

Se pide implementar el método **ciudadesAfectadas(String nomCiudad, float intensidad)**, que dada la ciudad epicentro del terremoto y la intensidad del mismo, devuelve una lista de las ciudades afectadas por el terremoto junto con la intensidad con la que les ha afectado.

Consideraremos que una ciudad ha sido afectada por el terremoto si la intensidad con la que le afecta es mayor o igual que 1.

En el grafo del ejemplo, la llamada **ciudadesAfectadas("L", 6.0)** devolvería que:

- L ha sido afectada con intensidad 6.0
- V, T y E han sido afectadas con intensidad 3.0
- S, A, W, C, Q, P, M han sido afectadas con intensidad 1.5

EJEMPLOS DE EJECUCIÓN DE LOS PROGRAMAS DE PRUEBA

Ejercicio 1 (se acepta cualquier camino que tenga la misma longitud y cumpla la condición)

```
### PRUEBA 1 ###

Camino más corto entre 11 y 20: [11, 13, 5, 9, 1, 17, 20]


### PRUEBA 2 ###

Camino más corto entre 4 y 13: [4, 3, 1, 9, 5, 13]


### PRUEBA 3 ###

Camino más corto entre 9 y 7: []


### PRUEBA 4 ###

Camino más corto entre 9 y 12: [9, 1, 17, 20, 12]
```

Ejercicio 2 **(el orden no es relevante)**

PRUEBA 1

Ciudades afectadas por un terremoto de intensidad 6 en L: [L: 6.0, V: 3.0, T: 3.0, E: 3.0, S: 1.5, A: 1.5, W: 1.5, C: 1.5, Q: 1.5, P: 1.5, M: 1.5]

PRUEBA 2

Ciudades afectadas por un terremoto de intensidad 2 en H: [H: 2.0, K: 1.0, A: 1.0]

PRUEBA 3

Ciudades afectadas por un terremoto de intensidad 8.5 en Q: [Q: 8.5, D: 4.25, E: 4.25, N: 4.25, L: 2.125, P: 2.125, M: 2.125, V: 1.0625, T: 1.0625, O: 1.0625, R: 1.0625]

PRUEBA 4

Ciudades afectadas por un terremoto de intensidad 0.8 en L: []