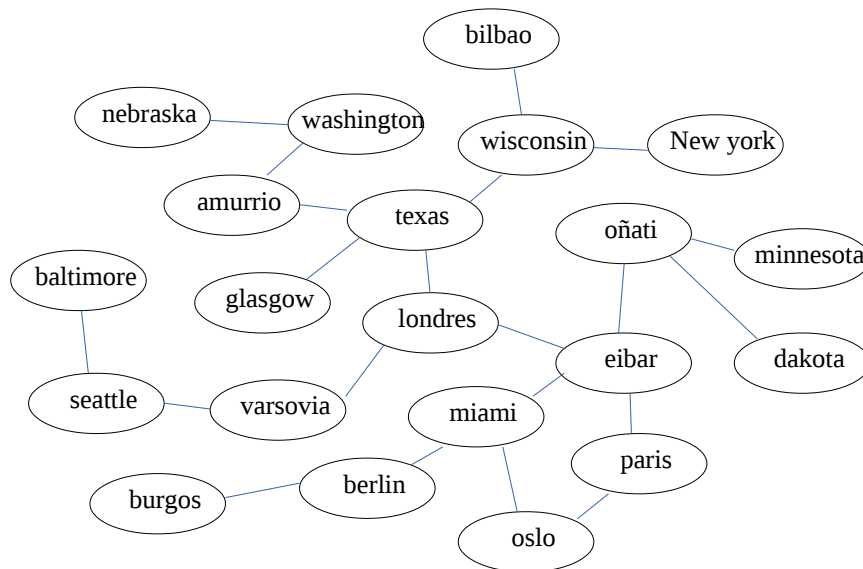


3. Terremoto (1,5 puntos)

Tenemos el siguiente grafo no dirigido, que representa comarcas. Un arco entre dos comarcas indica que esas dos comarcas tienen frontera en común, es decir, son adyacentes:



Cuando se produce un terremoto de escala N en una comarca, ese terremoto se propaga a las comarcas colindantes con una intensidad igual a la mitad de N , y así sucesivamente va perdiendo fuerza.

Se quiere conocer las comarcas afectadas por un terremoto de intensidad N . Se considera que una comarca es afectada por el terremoto si la intensidad en esa comarca es igual o superior a 1.

Por ejemplo, si se produjera un terremoto de intensidad 6 en *Londres*, afectaría a *texas*, *eibar* y *varsovia* con intensidad 3, y a *glasgow*, *amurrio*, *wisconsin*, *oñati*, *paris*, *miami*, y *seattle* con intensidad 1,5.

```
public class Graph
{
    protected final int DEFAULT_CAPACITY = 100;
    protected int numVertices; // number of vertices in the graph
    protected boolean[][] adjMatrix; // adjacency matrix
    protected String[] vertices; // values of vertices

    public int index(String t) { // ¡NO HAY QUE IMPLEMENTARLO!
        // pre: el elemento t se encuentra en el grafo
        // post devuelve el índice del array "vertices" correspondiente a t
    }

    public ArrayList<String> comarcasAfectadas(int intensidad, String c)
    // pre: "intensidad" indica la magnitud del terremoto
    // "c" es la comarca en que se ha producido el terremoto
    // post: el resultado es la lista de las comarcas afectadas por el terremoto
    // de esa intensidad en la comarca dada.
    // Una comarca estará afectada si el terremoto llega con intensidad igual
    // o superior a 1.
    // La intensidad del terremoto disminuye a la mitad al pasar de una comarca
    // a las comarcas limítrofes
}
```

Se pide:

1. Implementar el método "`comarcasAfectadas()`"
2. Calcular, de manera razonada, el coste del algoritmo resultante.