

Agenda - Grafos

- Recorrido en amplitud: BFS (Breath First Search)
- Ejemplo: El virus de computadora

Grafos

BFS (Breath First Search)

Este algoritmo es la generalización del recorrido por niveles de un árbol. La estrategia es la siguiente:

- Partir de algún vértice v , visitar v , después visitar cada uno de los vértices adyacentes a v .
- Repetir el proceso para cada nodo adyacente a v , siguiendo el orden en que fueron visitados.

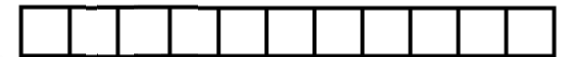
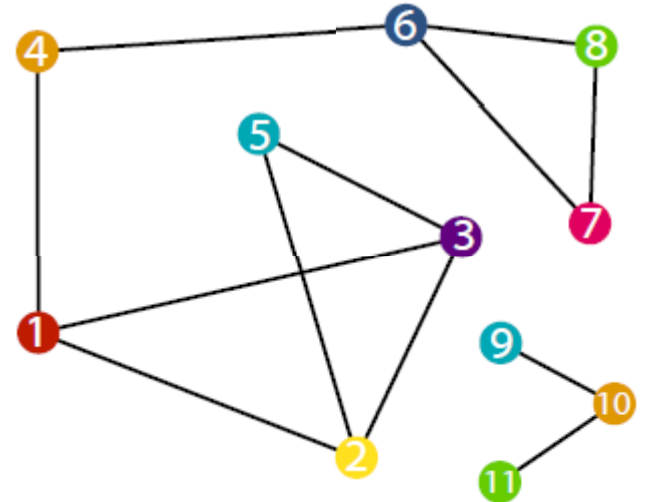
Si desde v no fueran alcanzables todos los nodos del grafo: elegir un nuevo vértice de partida no visitado, y repetir el proceso hasta que se hayan recorrido todos los vértices.

```
public class Recorridos {
    public void bfs(Grafo<T> grafo) {
        boolean[] marca = new boolean[grafo.listaDeVertices().tamano()];
        for (int i = 0; i < marca.length; i++) {
            if (!marca[i])
                this.bfs(i+1, grafo, marca); //las listas empiezan en la pos 1
        }
    }
    private void bfs (int i, Grafo<T> grafo, boolean[] marca) {
        //siguiente diapo
    }
}
```

Grafos

BFS (Breath First Search)

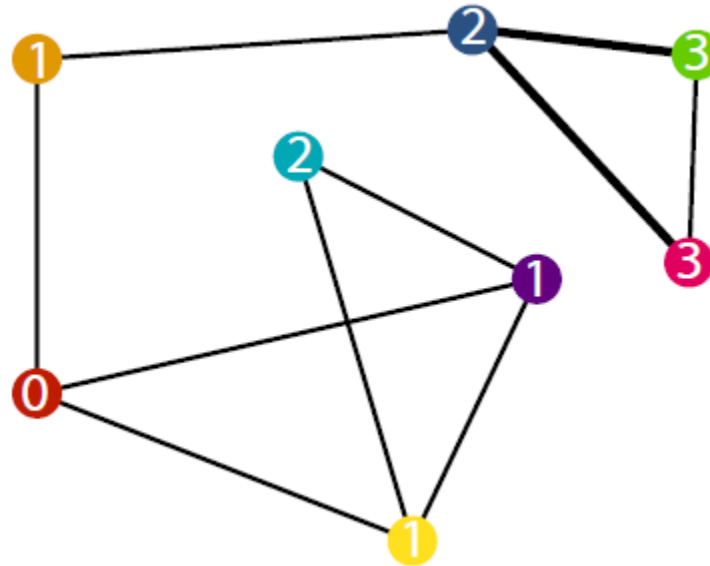
```
public class Recorridos {  
  
    private void bfs(int i, Grafo<T> grafo, boolean[] marca) {  
        ListaGenerica<Arista<T>> ady = null;  
        ColaGenerica<Vertice<T>> q = new ColaGenerica<Vertice<T>>();  
        q.encolar(grafo.listaDeVertices().elemento(i));  
        marca[i] = true;  
        while (!q.esVacia()) {  
            Vertice<T> v = q.desencolar();  
            System.out.println(v);  
            ady = grafo.listaDeAdyacentes(v);  
            ady.comenzar();  
            while (!ady.fin()) {  
                Arista<T> arista = ady.proximo();  
                int j = arista.getDestino().posicion();  
                if (!marca[j]) {  
                    Vertice<T> w = arista.getDestino();  
                    marca[j] = true;  
                    q.encolar(w);  
                }  
            }  
        }  
    }  
}
```



q

Ejemplo: El virus de computadora

Un poderoso e inteligente virus de computadora infecta cualquier computadora en 1 minuto, logrando infectar toda la red de una empresa con cientos de computadoras. Dado un grafo que representa las conexiones entre las computadoras de la empresa, y una computadora ya infectada, escriba un programa en Java que permita determinar el tiempo que demora el virus en infectar el resto de las computadoras. Asuma que todas las computadoras pueden ser infectadas, no todas las computadoras tienen conexión directa entre si, y un mismo virus puede infectar un grupo de computadoras al mismo tiempo sin importar la cantidad.



```

public class BFSVirus {
    public int calcularTiempoInfeccion(Grafo<String> g, Vertice<String> inicial) {
        int n = g.listaDeVertices().tamano();
        ColaGenerica<Vertice<String>> cola = new ColaGenerica<Vertice<String>>();
        int distancias[] = new int[n+1];    //no se usa la posicion 0
        int maxDist = 0; int nuevaDist = 0;
        for (int i = 0; i<=n; ++i) {
            distancias[i] = Integer.MAX_VALUE;
        }
        distancias[inicial.posicion()] = 0;
        cola.encolar(inicial);
        while (!cola.esVacia()) {
            Vertice<String> v = cola.desencolar();
            nuevaDist = distancias[v.posicion()] + 1;
            ListaGenerica<Arista<String>> adyacentes = v.obtenerAdyacentes();
            adyacentes.comenzar();
            while (!adyacentes.fin()) {
                Arista<String> a = adyacentes.proximo();
                Vertice<String> w = a.getDestino();
                int pos = w.posicion();
                if (distancias[pos]== Integer.MAX_VALUE) {
                    distancias[pos] = nuevaDist;
                    if (nuevaDist > maxDist)
                        maxDist = nuevaDist;
                    cola.encolar(w);
                }
            }
        }
        return maxDist;
    }
}

```

