

Agenda - Grafos

Recorrido en amplitud: BFS (Breath First Search)

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.

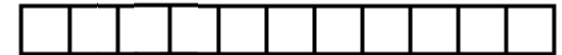
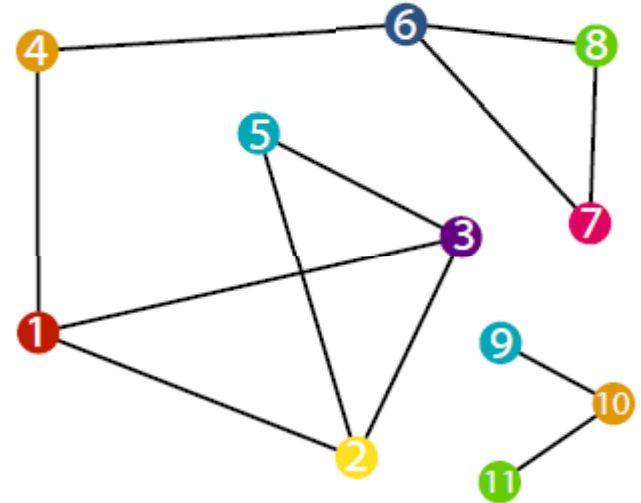
Si desde v no fueran alcanzables todos los nodos del grafo: elegir un nuevo vértice de partida no visitado, y repetir el proceso hasta que se hayan recorrido todos los vértices.

```
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