Ocampo Villegas Roberto 316293336
David Alvarado Torres 316167613

Computación Distribuida: Práctica 0

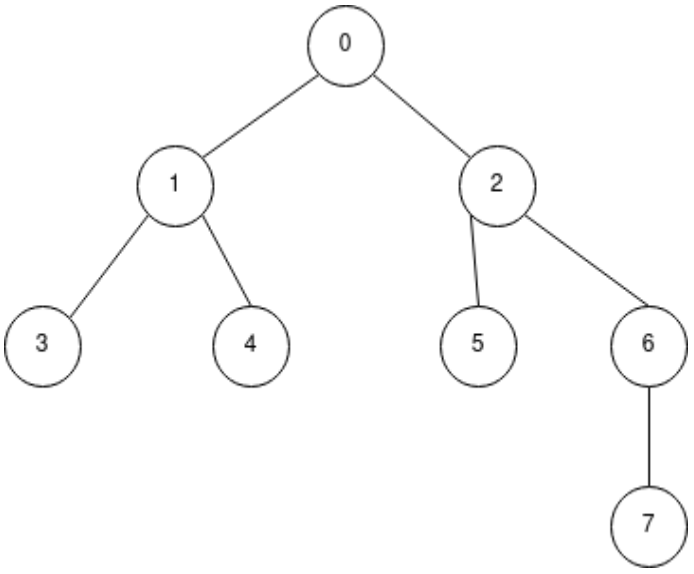
Para esta práctica creamos una clase gráfica, la cual tiene dos variables llamadas numeroVertices y listaVertices, la primera contiene el numero de vertices con el que cuenta la gráfica y la segunda la usaremos como una lista de listas, en donde cada elemento de listaVertices[i] representará los vecinos del i-ésimo nodo. Esta clase también va a contener un método agregarArista() el cual va a unir a dos vértices de la gráfica.

A BFS lo implementamos con tres listas, una que contiene la lista de nodos recorridos (su variable se llama listaFinal), otra que utilizamos para saber qué nodos ya han sido visitados (su variable se llama visitados), y otra que utilizaremos como la cola (su variable se llama cola). También recibe de argumento la gráfica a recorrer y el número de vértice de donde empezará a ser recorrida (variable n).

Los pasos del algoritmo son:

1. Agregamos a visitados una cantidad de 0 igual a la cantidad de vértices en la gráfica, 0 significa que no han sido visitados y 1 que ya.
2. Cambiamos a visitados[n] como 1 y lo agregamos a la cola.
3. Mientras la cola no esté vacía sacamos al primer elemento y lo agregamos a listaFinal.
4. Vemos la lista de vecinos de n y si estos no están marcados como visitado, entonces los agregamos a cola y los marcamos como visitados.
5. Se repite el paso 3 hasta que ya no haya nadie en la cola.
6. Se imprime listaFinal.

La gráfica de ejemplo viene en el código, pero es esta:



P.D. El método funciona para cualquier gráfica pero los árboles son un gran ejemplo del funcionamiento.