Practica No. 10

Algoritmo primero en anchura

Nombre(s):

César Eduardo Elias del Hoyo ID: 262045

Ximena Rivera Delgadillo ID: 261261

José Luis Sandoval Pérez ID: 261731

Objetivo:

Con la realización de esta práctica se pretende: que el alumno se familiarice con la implementación de un algoritmo que recorre en anchura a un grafo creando un árbol abarcador.

Fundamento Teórico:

Algoritmo de búsqueda en anchura

Es un algoritmo para recorrer o buscar elementos en un grafo (usado frecuentemente sobre árboles).

Intuitivamente, se comienza en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo) y se exploran todos los vecinos de este nodo.

A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el árbol.

Su nombre se debe a que expande uniformemente la frontera entre lo descubierto y lo no descubierto. Llega a los nodos de distancia *k*, sólo tras haber llegado a todos los nodos a distancia *k-1*.

Formalmente, **BFS** (proviene del inglés *Breadth-first search*) es un algoritmo de búsqueda sin información, que expande y examina todos los nodos de un árbol sistemáticamente para buscar una solución.

El algoritmo no usa ninguna estrategia heurística. El peso de los arcos para ejecutar BFS debe de ser de IGUAL costo.

Si los arcos tienen pesos negativos se aplica el algoritmo de Bellman-Ford en alguna de sus dos versiones.

Descripción detallada:

- Dado un nodo fuente *s*, se explora los nodos de *D* para *descubrir* todos los nodos alcanzables desde *s*.
- Se busca desde *s* a todos los nodos alcanzables.
- Después produce un árbol **BF** con raíz en *s* y que contiene a todos los nodos alcanzables.

- El camino desde *s* a cada nodo en este recorrido contiene el mínimo número de nodos. Es el camino más corto medido en número de nodos.

Durante un recorrido en anchura, cuando se recorren ciertos arcos, llevan a nodos sin visitar.

Los arcos que llevan a nodos nuevos se conocen como arcos de árbol y forman un bosque abarcador en anchura para el grafo dirigido dado.

Además de los arcos de árbol, existen dos tipos de arcos definidos por una búsqueda en anchura de un grafo dirigido, que se conocen como:

- **Arco de retroceso**: Es el arco que va de un nodo a uno de sus antecesores. Un arco que va de un nodo hacia si mismo se considera un arco de retroceso.
- **Arco cruzado**: Es el arco que va de un nodo a otro que no es ni antecesor ni descendiente.

Forma de trabajo:

Colaborativa en equipos de 3 personas

Material:

1. Computadora
2. IDE ANSI C

Procedimiento:

Se va a crear un programa que implemente el algoritmo primero en anchura.
A continuación, se muestra el seudocódigo:

```
BFS(grafo G, nodo_fuente s)
{
   // recorremos todos los vértices del grafo inicializándolos a NO_VISITADO,
   // distancia INFINITA y padre de cada nodo NULL
   for u ∈ V[G] do
   {
      estado[u] = NO_VISITADO;
      distancia[u] = INFINITO; /* distancia infinita si el nodo no es alcanzable */
      padre[u] = NULL;
   }
   estado[s] = VISITADO;
   distancia[s] = 0;
   Encolar(Q, s);
   while Q != 0 do
   {
      // extraemos el nodo u de la cola Q y exploramos todos sus nodos adyacentes
      u = extraer(Q);
      for  v ∈ adyacencia[u]  do
      {
         if estado[v] == NO_VISITADO then
         {
              estado[v] = VISITADO;
              distancia[v] = distancia[u] + 1;
              padre[v] = u;
              Encolar(Q, v);
         }
      }
   }
}
```

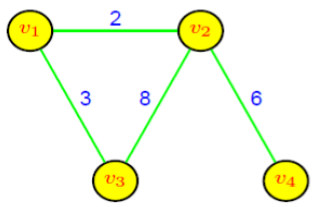

Figura 1

Para la creación del programa deberán realizarse los siguientes pasos:

1. En las primeras líneas elaborar comentarios con la siguiente información:
   a. Nombre de la institución
   b. Nombre de la carrera
   c. Nombre de la materia
   d. Nombre(s) de quien(es) realiza(n) la práctica
   e. Nombre del profesor
   f. Una descripción breve de lo que realiza el programa
2. Incluir las librerías necesarias.
3. Crear la matriz de adyacencia y desplegarla en pantalla.
4. Implementar el algoritmo búsqueda en profundidad y obtener el árbol abarcador de la figura 1 (seguir un orden numérico creciente).
5. Desplegando la matriz del árbol resultante.
6. Al salir se debe detener el programa y luego regresar el control al sistema inicial.

Resultados:

Realizar al menos dos corridas de prueba y mostrar los resultados en las pantallas de árbol genarado

Recorrido en anchura

```
PASOS del recorrido en anchura
        El nodo (a) con un nivel [0]
        El nodo (b) con un nivel [1]
        El nodo (c) con un nivel [1]
        El nodo (d) con un nivel [2]
```

```
MATRIZ del arbol resultante
        0    2    3    0
        0    0    0    6
        0    0    0    0
        0    0    0    0
```

Conclusiones:

En esta práctica revisamos el recorrido y búsqueda en anchura el cual no permite tener un método de búsqueda y fácil obtención de un valor dentro de un árbol o grafo. Es de mucha utilidad este algoritmo porque nos permite tener una rápida y eficaz obtención del elemento a buscar, recorriendo cada uno de los nodos del grafo seleccionado. Existen varios métodos de búsqueda y recorrido, pero sin duda el algoritmo empleado en esta práctica es uno de los mejores.