
Universidad de San Carlos de Guatemala

Matemática para Computación 2
Sección A

Facultad de ingeniería

Ing. José Alfredo González Díaz

Escuela de ciencias

Aux. Edgar Daniel Cil Peñate

Departamento de matemática

Segundo semestre 2023

Nombre: **Douglas Xavier Santiago Soto Mejia**

Carné: **202201502**



Proyecto

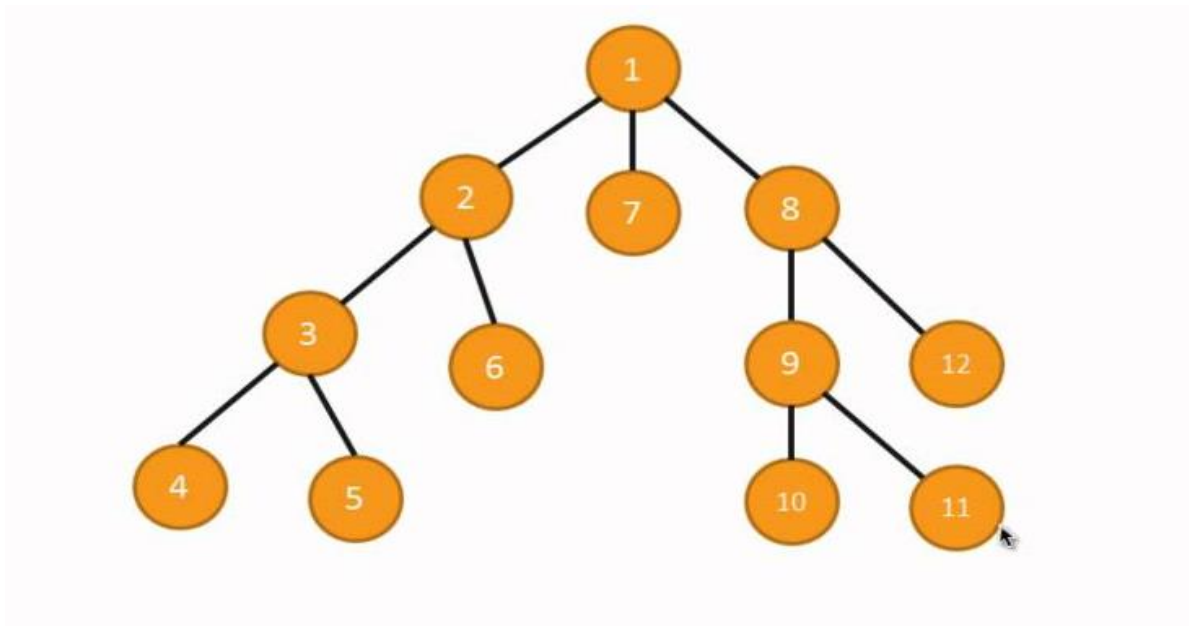
Algoritmo de búsqueda en los árboles

Parte de la realización de grafos, contiene el tema de lo que conocemos como árboles, y entre estos, se tiene un método de realización para los mismos, los cuales se conocen como los algoritmos de búsqueda a lo largo y a lo ancho. Los cuales en las matemáticas discretas y otras ramas de la ingeniería en ciencias y sistema tienen diversas aplicaciones, la cuales se verán más adelante.

Algoritmo de búsqueda a lo largo

Para el árbol de búsqueda a lo largo, se tiene un determinado grafo G , para realizar el mismo se tiene que, para empezar, si no hay un orden especificado, se debe hacer en orden ascendente (de menor a mayor), se toma el vértice de índice más pequeño y se debe dirigir al vértice de menor índice más cercano, se debe realizar el proceso tomando en cuenta de que, al ser un árbol, este no debe formar ciclos.

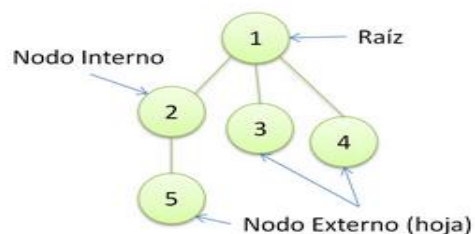
Una vez terminado el proceso de unir todas las aristas, lo que debemos realizar a continuación es la realización del árbol, tomando el primer vértice como raíz de este, en otras palabras, el punto de partida.



Algoritmo de búsqueda a lo ancho

Para la búsqueda a lo ancho el fin es el mismo, conseguir un árbol sin ciclos, la diferencia es que, al tomar el vértice, las aristas se despliegan hacia todos los vértices que sean alcanzables, se debe hacer lo mismo para todos los vértices, nuevamente tomando en cuenta no formar ciclos.

Y de la misma forma que en la búsqueda a lo largo, luego de realizar la búsqueda de nodos, se debe realizar el árbol correspondiente a dicho grafo, nuevamente comenzando por el vértice raíz.



Para ambos tipos de búsqueda de algoritmos se tienen dos opciones para el conteo de los nodos, orden ascendente y orden descendente. Los algoritmos tienen como regla que, el orden por el que se debe comenzar a tomar los nodos debe ser ascendente, pero solo en caso de que no sea especificado en el enunciado del problema a resolver.

Aplicaciones

Los usos de los algoritmos tienen sus aplicaciones, los más destacados son sus usos a las ciencias de la computación, los cuales son los siguientes:

Manejo de Datos:

La estructura que se tiene entre la nube y los servidores es la de un árbol, la nube funciona como vértice raíz mientras los servidores como los ordenadores, teléfonos funcionan como los nodos hijos que surgen de la raíz.

En los videojuegos:

Con el fin de que los tiempos de ejecución y el uso de la memoria sea lo más eficiente posible, se quiere llegar a la resolución de un problema común en el punto de vista informático de un videojuego, que es el desarrollo e inteligencia artificial. El camino para que esto se lleve a cabo es el desarrollo de algoritmos de búsqueda para la optimización de las diversas funciones que contiene dicha inteligencia artificial de un videojuego.

Registro de Datos:

Cuando necesitamos un campo en donde ingresar datos, pero en cada determinado momento su campo llave es almacenado en un índice, en el cual esas llaves están ordenadas de menor a mayor o de mayor a menor dependiendo el uso que se le dé. De esta manera, para agilizar la búsqueda de un registro en particular se da el acceso a ese registro por medio de su campo llave almacenado en el índice.

Un ejemplo de nuestra vida diaria y donde se aplica esta búsqueda es en un negocio, ya que aquí se necesita una búsqueda eficiente con una sola clave de acceso y otorgándonos la información requerida.

Implementaciones en programación

Python:

Cuando se tiene un algoritmo de búsqueda, este se basa en la propiedad de que las claves que son menores que el padre se encuentran en el subárbol izquierdo, y las claves que son mayores que el padre se encuentran en el subárbol derecho. A esto se le puede llamar abb. A medida que implementemos la interfaz de Vector Asociativo como se describió anteriormente, la propiedad abb guiará nuestra implementación.

```
class ArbolBinarioBusqueda:

    def __init__(self):
        self.raiz = None
        self.tamano = 0

    def longitud(self):
        return self.tamano

    def __len__(self):
        return self.tamano

    def __iter__(self):
        return self.raiz.__iter__()
```

C++:

Para el caso de c++ lo que se debe hacer al crear una estructura para el algoritmo de anchura que permita el almacenamiento de los nodos en un nivel.

```
queue<State> q;
q.push(State(raiz));
mark[raiz] = true;
```

posteriormente se declara la estructura de cola, agregando también el nodo raíz y así comenzar con el proceso de búsqueda.

```

while(!q.empty())
{
    State st = q.front();
    q.pop();
    if (st.node == nodo){
        printf("%d\n",nodo);
        return;
    }else printf("%d ",st.node);

    int T = (int)graph.G[st.node].size();
    for(int i = 0; i < T; ++i)
    {
        if (!mark[graph.G[st.node][i].node])
        {
            mark[graph.G[st.node][i].node] = true;
            q.push(State(graph.G[st.node][i].node));
        }
    }
}

```

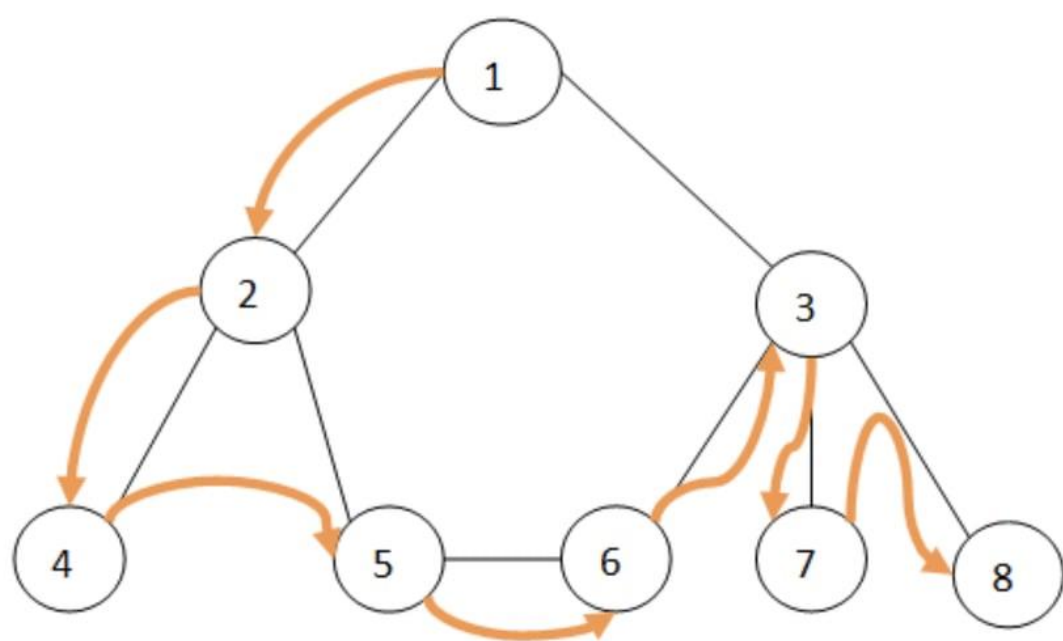
Opiniones del tema

El tema al poder ser utilizado en grafos y también teniendo la posibilidad de ser utilizado para determinar si un grafo es o no bipartito, tiene la utilidad de ser aplicado a la resolución de problemas de optimización. Además de que puede ser utilizado para recorrer un grafo de la forma más eficiente o corta posible.

- Por Bárbara Delgado.

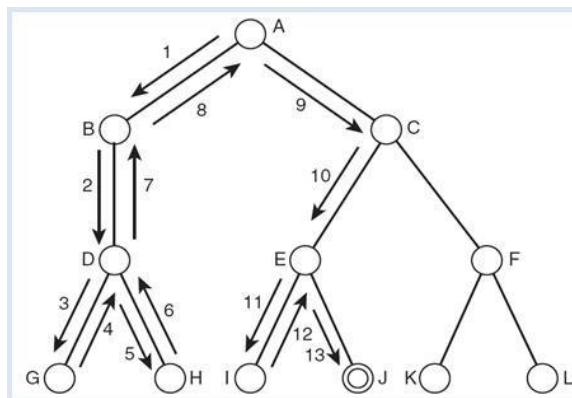
Para el análisis de una solución de una estructura en ramas como las ciencias y sistemas, es de suma importancia tener conocimiento de algoritmos específicos, tales como los algoritmos de recorridos de nodos, dos de ellos siendo de los más importantes son, el Algoritmo de búsqueda en profundidad y el Algoritmo de búsqueda en anchura. Sus siglas en ingles son: "Depth first search" y "Breadth first search" respectivamente.

- Por Josué Murillo



Conclusiones

- El tema de búsqueda por este tipo de algoritmos es uno de los procesos más eficientes en el momento de querer recorrer un grafo con la finalidad de conseguir un árbol que surge a partir de una raíz.
- Los Algoritmos de búsqueda tienen además de lo ya mencionado, distintas aplicaciones que pueden ser integradas a otros temas, principalmente a las ciencias de la computación, donde entra en juego la carrera de ciencias y sistemas en donde el tema puede ser aplicado constantemente, un ejemplo es en las matemáticas discretas.
- Ambos algoritmos tienen sus ventajas y desventajas, cada persona al saberlas puede distinguir con más facilidad cuál de los dos será el más eficiente para la búsqueda de los respectivos árboles.
- Ingenieros con experiencia y conocimiento del tema saben de la importancia de estos algoritmos ya que constantemente recuerdan la relevancia de los mismos a la hora de explorar los grafos con el fin de encontrar árboles, además de que cuando se toca el extenso tema de los grafos, los algoritmos de búsqueda entran en enseñanza.
- La implementación en distintos lenguajes de programación es posible, y puede ser en distintos lenguajes tomando en cuenta la finalidad de los códigos fuentes, que es la búsqueda por anchura y por profundidad.



Bibliografía

(pythoned, s.f., pág.

<https://runestone.academy/ns/books/published/pythoned/Trees/ImplementacionArbolBusqueda.html>)

(bibliadelprogramador, s.f., págs.

<https://www.bibliadelprogramador.com/2014/04/algoritmos-de-busqueda-en-anchura-bfs-y.html>)

(encora, s.f., págs. <https://www.encora.com/es/blog/dfs-vs-bfs>)