

Fecha: **02/11/2023**

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: **Jose Daniel Guzman Hernandez** Carné: **202111612**

Proyecto#1

Búsqueda en Amplitud (Búsqueda por Ancho):

Este algoritmo explora el árbol nivel por nivel, visitando todos los nodos de un nivel antes de pasar al siguiente. Utiliza una cola para mantener un registro de los nodos a visitar.

$O(V + E)$ donde V es el número de vértices y E es el número de aristas.

Aplicaciones:

Resolución de puzzles (como el rompecabezas de 8 piezas).

Encontrar el camino más corto en un grafo no ponderado.

Búsqueda de anchura en redes sociales para encontrar conexiones.

Búsqueda en Profundidad (DFS):

Este algoritmo explora el árbol tan profundamente como sea posible antes de retroceder y probar una ruta diferente. Utiliza una pila para mantener un registro de los nodos a visitar.

$O(V + E)$ donde V es el número de vértices y E es el número de aristas.

Aplicaciones:

Encontrar componentes conectados en un grafo.

Solucionar laberintos.

En análisis sintáctico en compiladores.

Algoritmo para Árboles de Notación Polaca:

investigar y realizar un algoritmo para árboles de notación polaca, se necesita especificar el tipo de árbol de notación polaca que se está considerando (por ejemplo, árbol de expresión en notación polaca inversa) y las operaciones que se quieren realizar en él.

Supongamos que estamos trabajando con un árbol de expresión en notación polaca inversa y queremos evaluar la expresión. Para ello, se puede realizar lo siguiente en Jupyter Notebook:

Implementar una clase para los nodos del árbol.

Implementar una función para construir el árbol a partir de una expresión en notación polaca inversa.

Implementar una función para evaluar la expresión representada por el árbol.

Capturas de pantalla de la solución con código: (Se deberá agregar posteriormente tras realizar la implementación en Jupyter).

Conclusiones

Versatilidad: Los algoritmos de búsqueda en árboles son herramientas versátiles y fundamentales en ciencias de la computación, aplicables a una amplia variedad de problemas.

Eficiencia: Mientras ambos algoritmos, búsqueda en amplitud y búsqueda en profundidad, tienen una complejidad de $O(V + E)$, su eficiencia puede variar significativamente dependiendo del tipo de problema y la estructura de los datos.

Aplicabilidad: La elección entre búsqueda en amplitud y búsqueda en profundidad debe basarse en las necesidades específicas del problema a resolver, ya que cada algoritmo tiene sus propias fortalezas y debilidades.

Importancia de las Entrevistas y Encuestas: Las opiniones de expertos y profesionales que trabajan con estos algoritmos son cruciales para entender sus aplicaciones prácticas y limitaciones.

Necesidad de Experimentación: Implementar y experimentar con los algoritmos en diferentes contextos y con diferentes conjuntos de datos es esencial para obtener una comprensión completa de su comportamiento y rendimiento.