



Uleam
UNIVERSIDAD LAICA
ELOY ALFARO DE MANABÍ

UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ “ULEAM”

CARRERA

TECNOLOGÍA DE LA INFORMACIÓN

REALIZAR

INVESTIGACIÓN

FECHA

02/05/2024

ESTUDIANTE

DOMINGUEZ ZAMBRANO MERLY MISHELLE

CEVALLOS BRAVO KAREN MISHELLE

MATERIA

INTELIGENCIA ARTIFICIAL

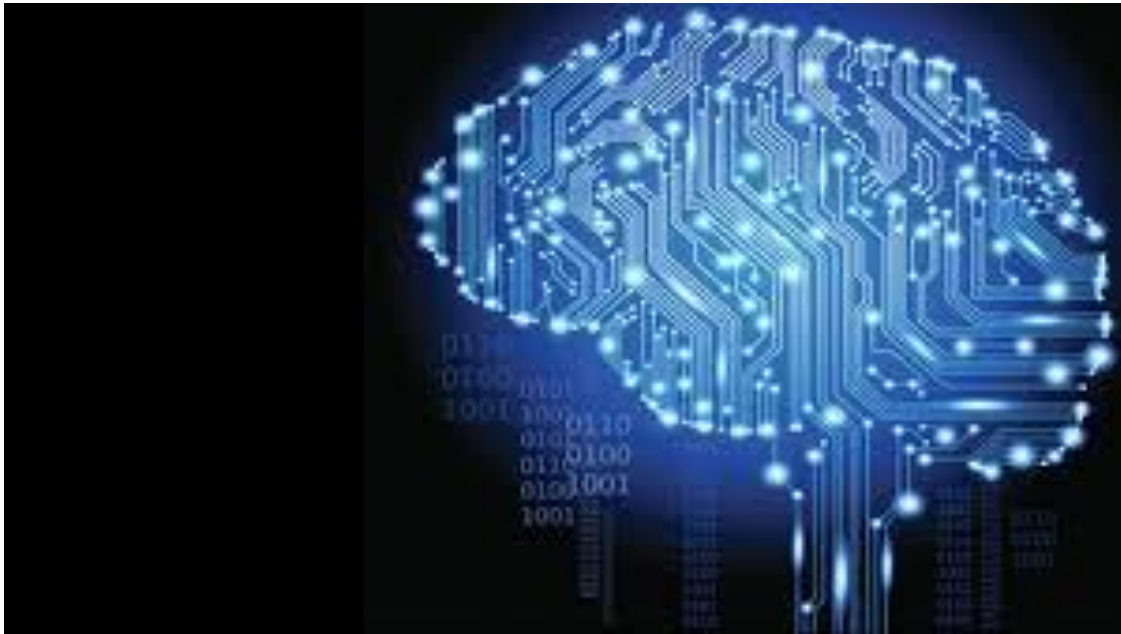
SEMESTRE

8^{vo} “A”

DOCENTE

ING. SINCHIGUANO CHIRIBOGA CESAR AUGUSTO

ALGORITMOS DE BUSQUEDA EN IA



➤ ¿Qué es algoritmo de búsqueda?

Un algoritmo de búsqueda es un conjunto de instrucciones paso a paso diseñadas para encontrar una solución o un elemento específico dentro de un conjunto de datos. Estos algoritmos son ampliamente utilizados en informática para buscar información en bases de datos, en la web, en archivos, etc. Algunos ejemplos de algoritmos de búsqueda son la búsqueda binaria, la búsqueda lineal y la búsqueda de profundidad. Estos algoritmos pueden variar en complejidad y eficiencia dependiendo del tamaño de los datos y la naturaleza del problema a resolver

➤ Importancia de los algoritmos de búsqueda

Los algoritmos de búsqueda juegan un papel fundamental en el campo de la informática, ya que permiten encontrar de manera eficiente y rápida la información deseada en grandes conjuntos de datos. Algunas de las principales razones por las cuales los algoritmos de búsqueda son importantes son las siguientes:

1. **Eficiencia:** Los algoritmos de búsqueda ayudan a optimizar el tiempo de búsqueda de la información deseada, permitiendo encontrarla en un tiempo mucho menor que si se realizara de forma manual o sin un algoritmo específico.

2. **Accesibilidad:** Gracias a los algoritmos de búsqueda, es posible acceder de manera rápida y sencilla a la información necesaria, lo que facilita la toma de decisiones y la resolución de problemas de manera eficiente.

3. **Organización de datos:** Los algoritmos de búsqueda permiten organizar y estructurar los datos de forma eficiente, facilitando su acceso y utilización posterior.

4. **Optimización de recursos:** Al utilizar algoritmos de búsqueda, se optimizan los recursos necesarios para realizar la búsqueda, como el tiempo, la memoria y el procesamiento, lo que mejora el rendimiento del sistema.

5. Mejora la productividad: Al reducir el tiempo necesario para buscar información, los algoritmos de búsqueda permiten a los usuarios ser más productivos y eficientes en sus tareas diarias.

➤ Propiedades de los algoritmos de búsqueda

1. **Eficiencia:** Los algoritmos de búsqueda deben ser eficientes en términos de tiempo y espacio, es decir, deben ser capaces de encontrar la solución en un tiempo razonable y utilizando la menor cantidad de recursos posibles.

2. **Complejidad:** La complejidad de un algoritmo de búsqueda se refiere a la cantidad de recursos que consume para encontrar la solución. Esto puede medirse en términos de tiempo, espacio o ambos.

3. **Naturaleza del problema:** Algunos algoritmos de búsqueda son más adecuados para ciertos tipos de problemas que otros. Por ejemplo, un algoritmo de búsqueda en profundidad puede ser más eficaz para problemas donde la solución se encuentra en una rama específica del árbol de búsqueda.

4. **Optimalidad:** Algunos algoritmos de búsqueda garantizan encontrar la solución óptima, es decir, la solución que cumple con ciertos criterios de optimización, como minimizar el coste o maximizar la ganancia.

5. **Completeness:** La completitud de un algoritmo de búsqueda se refiere a su capacidad para encontrar la solución si esta existe. Algunos algoritmos son incompletos y pueden no encontrar la solución en todos los casos.

6. **Robustez:** Los algoritmos de búsqueda deben ser capaces de manejar diferentes condiciones y variables en el problema, como entradas desordenadas o datos faltantes, sin afectar su capacidad para encontrar la solución

➤ ¿Cómo funcionan los algoritmos de búsqueda?

Los algoritmos de búsqueda son procedimientos que permiten encontrar un elemento específico dentro de un conjunto de datos. funcionan comparando el elemento buscado con los elementos del conjunto de datos de manera sistemática hasta encontrarlo o determinar que no está presente. La eficiencia de un algoritmo de búsqueda se mide en términos de su complejidad temporal, que indica el tiempo que tarda en encontrar el elemento deseado en función del tamaño del conjunto de datos.

TIPOS DE ALGORITMOS DE BUSQUEDA

➤ Algoritmo de búsqueda desinformado

1. **Búsqueda en amplitud (BFS):** Este algoritmo expande todos los nodos a la misma profundidad antes de expandir los nodos de la siguiente profundidad. Es ideal para encontrar la solución más corta en un grafo sin pesos.

2. **Búsqueda en profundidad (DFS):** Este algoritmo explora la rama completa antes de retroceder. Puede quedar atrapado en ciclos si no se implementa correctamente, pero es útil para encontrar soluciones en grafos con muchos niveles.

3. **Búsqueda de costo uniforme:** Este algoritmo expande los nodos con menor costo total hasta el momento. Es útil cuando se tienen grafos con pesos en las aristas, pero puede ser costoso en términos de memoria y tiempo.

4. **Búsqueda en profundidad limitada (DLS):** Este algoritmo es similar a DFS, pero se limita a una cierta profundidad. Es útil cuando se necesita evitar el retroceso excesivo en grafos con profundidad desconocida.

5. **Búsqueda en haz (Beam Search):** Este algoritmo es una versión modificada de BFS que solo conserva un número fijo de los nodos más prometedores en cada nivel. Es útil para conservar recursos, pero puede perder soluciones óptimas.

6. **Búsqueda en salto:** Este algoritmo expande nodos aleatoriamente en lugar de en un orden determinado. Es útil para evitar quedar atrapado en ciclos pero puede no ser eficiente en términos de tiempo.

➤ Algoritmo de búsqueda informados

Los algoritmos de búsqueda informados son aquellos que utilizan información sobre el estado actual de un problema para guiar la exploración y encontrar una solución de manera más eficiente. Algunos ejemplos de algoritmos de búsqueda informados son:

1. **Búsqueda A*:** es un algoritmo que combina la búsqueda en anchura con una heurística que estima el costo mínimo para llegar al objetivo. A* expande los nodos con menor costo total primero, lo que lo hace más eficiente que la búsqueda en anchura.

2. **Búsqueda de costo uniforme:** este algoritmo selecciona el camino que tiene el menor costo total en cada paso. A diferencia de la búsqueda A*, la búsqueda de costo uniforme no utiliza heurísticas para guiar la exploración, lo que puede hacerla menos eficiente en problemas más complejos.

3. **Búsqueda de primera mejor elección:** en este algoritmo, se selecciona el nodo sucesor que tiene el menor valor de una función de evaluación en cada paso. La función de evaluación combina el costo del camino hasta el nodo y una heurística que estima el costo restante para llegar al objetivo.