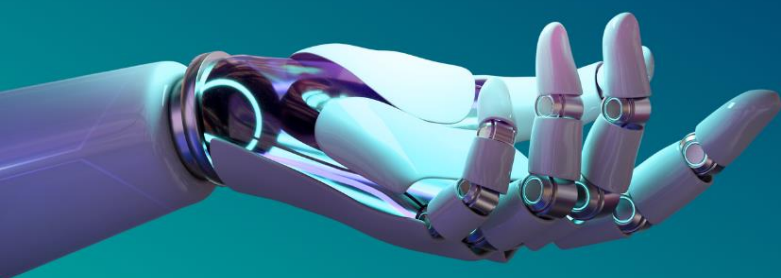


CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

**UNIVERSIDAD DE GUADALAJARA**

# TAREA 4

## ALGORITMOS DE BÚSQUEDA



### Alumnos:

Carbajal Armenta Yessenia Poala

Leon Estrada Rafael

Medina Bolaños Danna Paola

### Códigos:

220286482

220286121

220286938

### Maestro:

Oliva Navarro Diego Alberto

### Materia:

Inteligencia Artificial

### Sección:

D05

**Guadalajara, Jal. a 21 de abril del 2023**

---

## Algoritmos de búsqueda

---

### 1. Algoritmo de búsqueda en profundidad:

El algoritmo de búsqueda en profundidad es un algoritmo de búsqueda no informado que se utiliza para buscar en árboles o gráficos. Comienza explorando el nodo raíz y luego explora recursivamente todos los nodos hijos del nodo actual hasta que se alcanza el objetivo o se agota el espacio de búsqueda. Este algoritmo puede ser ineficiente para árboles o gráficos profundos y puede no encontrar soluciones óptimas. Una de las ventajas de este algoritmo es que utiliza menos memoria en comparación con otros algoritmos de búsqueda, pero la principal deficiencia es que puede caer en bucles infinitos en árboles con ciclos.

- **Ventajas:** utiliza menos memoria en comparación con otros algoritmos de búsqueda.
- **Deficiencias:** puede caer en bucles infinitos en árboles con ciclos, no garantiza encontrar la solución óptima.
- **Diferencias:** comienza explorando el nodo raíz y luego explora recursivamente todos los nodos hijos del nodo actual.

### 2. Algoritmo de búsqueda en amplitud:

El algoritmo de búsqueda en amplitud es otro algoritmo no informado utilizado para buscar en árboles o gráficos. Comienza explorando el nodo raíz y luego explora todos los nodos hermanos del nodo actual antes de pasar a los nodos hijos de esos hermanos. Esto continúa hasta que se alcanza el objetivo o se agota el espacio de búsqueda. A diferencia del algoritmo de búsqueda en profundidad, el algoritmo de búsqueda en amplitud garantiza encontrar la solución óptima. Sin embargo, este algoritmo puede ser ineficiente para árboles o gráficos profundos y utiliza más memoria que el algoritmo de búsqueda en profundidad.

- **Ventajas:** garantiza encontrar la solución óptima.
- **Deficiencias:** puede ser ineficiente para árboles

- **Diferencias:** explora todos los nodos del mismo nivel antes de avanzar al siguiente, mientras que DFS explora todos los nodos de un camino antes de retroceder al nivel anterior.

### 3. Algoritmos voraces:

Los algoritmos voraces son algoritmos informados que toman decisiones en función de una función heurística. Estos algoritmos seleccionan el nodo que parece ser el mejor candidato para acercarse al objetivo. El algoritmo voraz no garantiza encontrar la solución óptima, pero puede ser muy eficiente. Una deficiencia común de los algoritmos voraces es que pueden caer en mínimos locales y no encontrar la solución óptima.

- **Ventajas:** pueden ser muy eficientes.
- **Deficiencias:** pueden caer en mínimos locales y no encontrar la solución óptima.
- **Diferencias:** toman decisiones en función de una función heurística.

### 4. Algoritmo A\*:

El algoritmo A\* es un algoritmo informado que utiliza una combinación de la función heurística y el costo de movimiento para decidir cuál es el siguiente nodo a explorar. Este algoritmo es muy eficiente y garantiza encontrar la solución óptima si se utiliza una función heurística admisible. Sin embargo, este algoritmo puede ser ineficiente si la función heurística no es admisible o si el espacio de búsqueda es demasiado grande.

- **Ventajas:** es muy eficiente y garantiza encontrar la solución óptima si se utiliza una función heurística admisible.
- **Deficiencias:** puede ser ineficiente si la función heurística no es admisible o si el espacio de búsqueda es demasiado grande.
- **Diferencias:** utiliza una combinación de la función heurística y el costo de movimiento para decidir cuál es el siguiente nodo a explorar.

### Sugerencias para solventar las deficiencias:

- **Algoritmo de búsqueda en profundidad:** se puede implementar una comprobación de ciclo para evitar caer en bucles infinitos en árboles con ciclos. También se puede utilizar una búsqueda en profundidad limitada para limitar la profundidad máxima de la búsqueda.
- **Algoritmo de búsqueda en amplitud:** se puede implementar una búsqueda en profundidad iterativa para mejorar la eficiencia en árboles o gráficos profundos. También se puede utilizar una búsqueda bidireccional para explorar el espacio de búsqueda desde el nodo inicial y el nodo objetivo.
- **Algoritmos voraces:** se pueden utilizar diferentes funciones heurísticas para evitar caer en mínimos locales y mejorar la calidad de las soluciones.
- **Algoritmo A\*:** se puede utilizar una función heurística admisible para garantizar la optimización de la solución. También se pueden utilizar diferentes estrategias de poda para reducir el espacio de búsqueda y mejorar la eficiencia.

### Referencias bibliográficas

- Página web de Stanford Online: Artificial Intelligence (<https://online.stanford.edu/courses/see-all-ai-courses>): aquí puedes encontrar cursos gratuitos en línea sobre inteligencia artificial que cubren los fundamentos de los algoritmos de búsqueda y otros temas relacionados.
- Página web de Medium: Artificial Intelligence (<https://medium.com/topic/artificial-intelligence>): esta plataforma de publicación de contenido cuenta con numerosos artículos sobre inteligencia artificial, incluyendo explicaciones detalladas de los algoritmos de búsqueda.
- Página web de Towards Data Science: Artificial Intelligence (<https://towardsdatascience.com/tagged/artificial-intelligence>): aquí puedes encontrar artículos sobre inteligencia artificial, incluyendo explicaciones de los algoritmos de búsqueda y sus aplicaciones en problemas de la vida real.

- Video de Computerphile: A\* Search Algorithm (<https://www.youtube.com/watch?v=ySN5Wnu88nE>): este video explica el algoritmo A\* y su aplicación en la resolución de problemas de búsqueda.
- Video de Code Bullet: Evolution of A\* Pathfinding ([https://www.youtube.com/watch?v=MII74b-5\\_w4](https://www.youtube.com/watch?v=MII74b-5_w4)): este video muestra cómo se puede mejorar el algoritmo A\* a través de la evolución y la adaptación.
- Video de Tom Scott: The Difficulty of Faking Data with Machine Learning (<https://www.youtube.com/watch?v=V9O-1h-ORFY>): este video explica cómo los algoritmos de búsqueda y otros algoritmos de inteligencia artificial pueden ser utilizados para falsificar datos.