



# FACULTAD DE CIENCIAS

## INTELIGENCIA ARTIFICIAL

---

### **Agentes inteligentes en laberintos**

---

Equipo: Skynet Scribes

Número de practica: 02

**Carlos Daniel Cortés Jiménez**  
420004846

**Sarah Sophía Olivares García**  
318360638

**Marco Silva Huerta**  
316205326

**Juan Daniel Barrera Holan**  
417079372

**Laura Itzel Tinoco Miguel**  
316020189

Profesora: Cecilia Reyes Peña

Ayudante teoría: Karem Ramos Calpulalpan

Ayudante laboratorio: Tania Michelle Rubí Rojas

**Semestre 2024-2**

Fecha de entrega:  
**21 de Febrero del 2024**

# 1. Backtracking

Backtracking es una técnica algorítmica para hacer una búsqueda exhaustiva y sistemática por todas las configuraciones posibles del espacio de búsqueda del problema, para encontrar el resultado definido por el problema. El término *backtrack* fue acuñado por primera vez por el matemático estadounidense D. H. Lehmer en la década de 1950.

En general, la forma de actuar consiste en elegir una alternativa del conjunto de opciones en cada etapa del proceso de resolución, y si esta elección no funciona (no nos lleva a ninguna solución), la búsqueda vuelve al punto donde se realizó esa elección, e intenta con otro valor. Cuando se han agotado todos los posibles valores en ese punto, la búsqueda vuelve a la anterior fase en la que se hizo otra elección entre valores. Si no hay más puntos de elección, la búsqueda finaliza.

Se suele aplicar en la resolución de un gran número de problemas, muy especialmente en los de decisión y optimización:

- **Problemas de decisión:** Búsqueda de las soluciones que satisfacen ciertas restricciones.  
*Ejemplo:* Problema de las N-Reinas.
- **Problemas de optimización:** Búsqueda de la mejor solución en base a una función objetivo.  
*Ejemplo:* Problema de la mochila.

Sus ventajas y desventajas son:

- **Ventajas:**
  - Si existe una solución, la calcula.
  - Es un esquema sencillo de implementar.
  - Adaptable a las características específicas de cada problema.
- **Desventajas:**
  - Coste exponencial en la mayoría de los casos.
  - Si el espacio de búsqueda es infinito, la solución, aunque exista, no se encontrará nunca.
  - Por término medio consume mucha memoria al tener que almacenar las llamadas recursivas.

# 2. Algoritmos similares

Algunos algoritmos que utilizan estrategias similares a Backtracking son:

- **Branch and Bound:** Este realiza un recorrido sistemático del árbol de estados de un problema, si bien ese recorrido no tiene por qué ser en profundidad, como sucedía en backtracking: usaremos una estrategia de ramificación. Aquí se utilizan técnicas de poda para poder eliminar aquellos nodos que no lleven a soluciones óptimas.
- **Depth-First Search (DFS):** Este es un algoritmo de búsqueda no informada, lo que hace primero es proporcionar los pasos para atravesar todos los nodos de un gráfico sin repetir ningún nodo.

- **Breadth-First Search (BFS):** Es un algoritmo para atravesar o buscar estructuras de datos de árboles. Comienza en la raíz del árbol (o algún nodo arbitrario, en ocasiones denominado como: clave de búsqueda) y explora primero los nodos vecinos antes de pasar a los vecinos del siguiente nivel.
- **A\* (A estrella):** A\* combina la búsqueda heurística con la búsqueda informada para encontrar la solución más eficiente en problemas de ruta. Aunque difiere del backtracking, comparten el objetivo de encontrar soluciones en un espacio de búsqueda.

### 3. Pseudocódigo

# aquí va el código

### 4. Documentación

## Referencias

- [Fig12] Jhosimar George Arias Figueroa. *Algorithms and More*. <https://jariasf.wordpress.com/2012/03/02/algoritmo-de-busqueda-depth-first-search-parte-1/>. 2012.
- [RN16] Stuart Russell y Peter Norvig. *Inteligencia Artificial Un Enfoque moderno*. 2nd. Pearson Prentice Hall, 2016.
- [Ref21] Programación Refactoriza. *Backtracking*. <https://docs.jjpeleato.com/algoritmia/backtracking>. 2021.