

¿Qué es la IA?

La inteligencia artificial o IA es, dentro de la informática, un campo de estudio encargado de desarrollar programas o algoritmos capaces de replicar operaciones atribuidas a la inteligencia humana.

Historia de la IA

En 1950 Alan Turing asienta las bases de la informática actual, consolida el campo de la inteligencia artificial y crea el **test de Turing**, que determina si una maquina es considerada inteligente o no.

En 1956, en la conferencia de Darmouth, se acuña el termino **Inteligencia Artificial**.

En 1957, Frank Rosenblat diseña la primera red neuronal artificial.

En 1966, el MIT desarrolla uno de los primeros programas (ELIZA) en procesar lenguaje natural y conversar a través de una serie de frases programadas.

EN 1979, un ordenador gana al campeón mundial de backgammon, lo que impulsa nuevos proyectos para juegos más complejos.

En 1981 Japón comienza el proyecto “Quinta Generación”, en el cual, este país invierte en una nueva generación de ordenadores que utilicen inteligencia artificial.

En 1987, **Martin Fischles** y **Oscar Firschein** describen atributos de un agente inteligente, lo cual abre nuevas áreas de investigación para la IA.

En 1996, la supercomputadora **Deep Blue** vence a Kasparov en una partida de ajedrez.

En 2005, usando la **Ley de Moore**, se predijo que las maquinas alcanzaran un nivel de inteligencia humano en 2029

En 2012, **Google** crea un superordenador capaz de aprender a través de YouTube a identificar caras y cuerpos humanos.

En 2014, un ordenador supera con éxito el **Test de Turing**.

En 2016, **Microsoft** lanza **Tal**, un chatbot capaz de aprender a partir de la interacción con las personas. Ese mismo año un ordenador de **Google** venció al campeón del juego “Go”.

En 2017, un software se impone ante rivales humanos en un torneo de póker.

¿Qué es la Teoría de Juegos?

La **Teoría de Juegos** es el estudio matemático en el cual se toman decisiones teniendo en cuenta las elecciones que hacen otros. Esta teoría ha contribuido en la comprensión de la conducta humana frente a la toma de decisiones.

Complejidad

Propiedad de un sistema lógico por la que cualquier expresión cerrada es derivable o refutable dentro del mismo sistema.

Complejidad temporal

En informática, es la propiedad computacional que define la eficiencia de un algoritmo, es decir, el tiempo que tarda y el número de operaciones óptimas que realiza un algoritmo.

Complejidad espacial

Se trata de la cantidad de memoria que necesita un algoritmo para llegar a completarse.

Optimidad

Mejor manera de realizar algo para llegar a resultado correcto.

Tipos de búsquedas

Búsqueda en anchura:

- Siempre encontrará la solución y además será óptima.
- Consume más recursos por la alta complejidad espacial y temporal.

Búsqueda en profundidad:

- Menor complejidad espacial
- No siempre encontrará la solución o encontrará una solución muy alejada si esta sin acotar
- Posibilidad de que aparezcan bucles infinitos

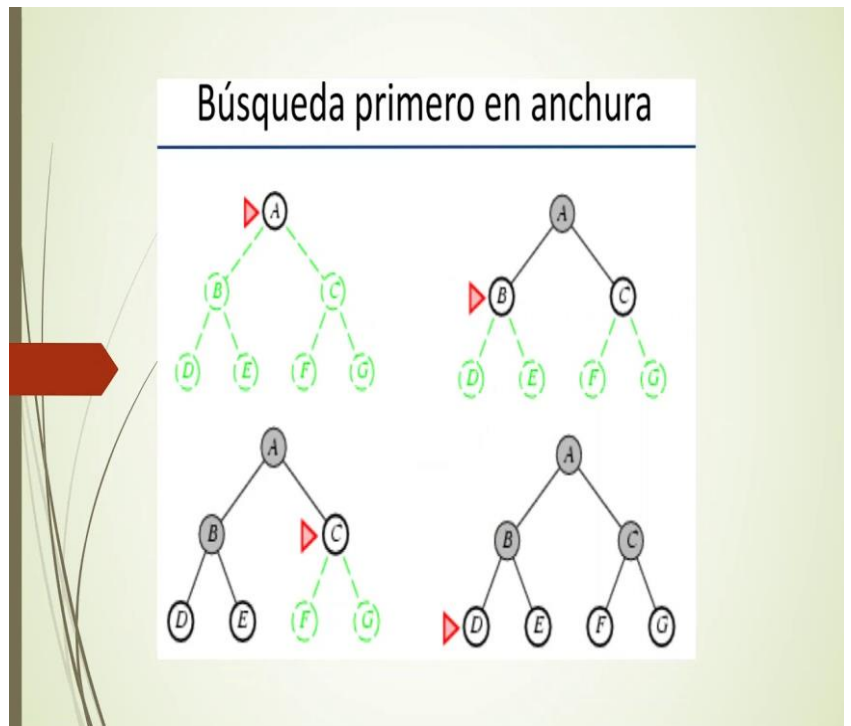
Búsqueda A*:

- Siempre encontrará la solución que sea más óptima y de menor coste
- Complejidad espacial elevada.

Búsqueda en anchura

Procede a buscar el recorrido nivel a nivel, es decir, no expande ningún nodo antes de haber expandido todos los del nivel anterior. Se implementa con una estructura FIFO (First in, first out).

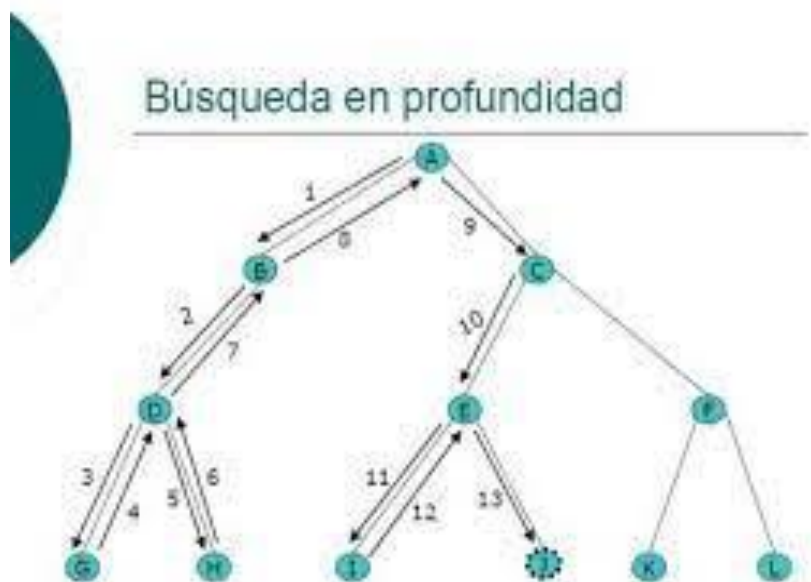
Este tipo de búsqueda asegura que, de haber solución, será una solución óptima.



Búsqueda en profundidad

La búsqueda se realiza por una sola rama del árbol hasta encontrar una solución o hasta que se tome la decisión de terminar la búsqueda por esa dirección. Si esto ocurre se produce una vuelta atrás y se sigue por otra rama hasta visitar todas las ramas del árbol si es necesario a menos que le indique un rango de nodos maximos a visitar.

Este tipo de busqueda es de menor complejidad que la de anchura, pero la solución podría no ser encontrada o que este muy alejada en cuanto a recorrido.



Búsqueda A*

Se trata de un método de búsqueda de tipo tentativo, que pondera a la vez lo cerca que estamos del nodo meta y lo lejos que estamos del nodo inicial.

Este tipo de búsqueda encontrara siempre el resultado que tenga el camino con el menor coste entre inicio y meta.

7	6	5	6	7	8	9	10	11		19	20	21	22
6	5	4	5	6	7	8	9	10		18	19	20	21
5	4	3	4	5	6	7	8	9		17	18	19	20
4	3	2	3	4	5	6	7	8		16	17	18	19
3	2	1	2	3	4	5	6	7		15	16	17	18
2	1	0	1	2	3	4	5	6		14	15	16	17
3	2	1	2	3	4	5	6	7		13	14	15	16
4	3	2	3	4	5	6	7	8		12	13	14	15
5	4	3	4	5	6	7	8	9	10	11	12	13	14
6	5	4	5	6	7	8	9	10	11	12	13	14	15

Explicación de juegos minimax y poda alfa-beta.

Un algoritmo minimax es un método de decisión para minimizar la pérdida máxima en un juego con un adversario y con toda la información. Ejemplos donde se podría aplicar este algoritmo serían juegos como el ajedrez, Go o las damas, juegos de dos jugadores, por turnos, donde ambos jugadores saben que posibles movimientos se podrían hacer, suponiendo siempre que el contrincante elegirá el peor movimiento para ti y viceversa.

Para reducir el coste computacional de minimax, se usa un método de poda alfa-beta que nos lleva a una solución peor a las ya encontradas. Con este método decidimos que ramas no serán exploradas, ahorrando espacio y tiempo computacional.

Los valores alfa son los calculados al ir hacia atrás en los nodos max. Estos valores no pueden crecer.

Los valores beta son, al contrario, son los calculados hacia atrás en los nodos min y no pueden crecer.