



Algoritmo A^{*}



Definición

- A* es un algoritmo de búsqueda **informada** que nos ayuda a encontrar el **camino más corto** (si es que existe) entre un nodo inicial y un nodo objetivo.
- Para encontrar el camino más corto, hace uso de una heurística (nosotros la elegimos).
- Es utilizado para resolver una gran variedad de problemas de búsqueda (mapas, videojuegos)

¿Cómo funciona?

- A* utiliza un **conjunto abierto** de nodos a explorar. Inicialmente solo contiene al nodo inicial (la entrada del laberinto).
- A cada nodo en el **conjunto abierto** se le asigna un valor **f** que es la suma de dos componentes:

$$f(n) = g(n) + h(n)$$

donde **g(n)**: es el costo *real* del movimiento desde el inicio hasta el nodo *n*, y **h(n)**: es el costo *estimado* del movimiento desde el nodo *n* hasta la salida (este valor está dado por la heurística).

¿Cómo funciona?

- A* elige al nodo cuyo valor **f** sea el menor para dar su siguiente movimiento.
- Recorremos toda la gráfica guiándonos por los valores de **f** hasta encontrar el nodo objetivo.
- ★ El resultado de nuestro recorrido depende de la heurística que utilicemos

La heurística que usaremos
para esta práctica será la
función de **la distancia**
Manhattan

Dados dos puntos (x,y) y (w,z) ,
la distancia se calcula como

$$|x - w| + |y - z|$$

Propiedades

- Es completo: siempre encontrará la solución en caso de que exista
- Si para toda n de la gráfica se cumple que $g(n) = 0$, entonces la búsqueda es igual a la de un algoritmo voraz (greedy).
- Si la heurística $h(n)$ es admisible, entonces A^* encuentra la mejor solución posible.

Consejos de implementación

- Entiende muy bien cómo se calcula la función de costo $f(n) = g(n) + h(n)$, pues esta función nos ayuda a determinar cuál será nuestro siguiente movimiento.
- Recuerda que la heurística que utilizaremos es la función para calcular la **distancia Manhattan**.
- Se recomienda usar un heap para implementar el conjunto abierto.

Consejos de implementación

- Se recomienda usar un conjunto o diccionario para implementar el conjunto cerrado.
- Para poder reconstruir la ruta, ve guardando el nodo anterior de cada nodo visitado.
- Documenta tu código
- Realiza varias pruebas para determinar si tu algoritmo funciona o no.