

# INTELIGENCIA ARTIFICIAL

## Algoritmos de búsqueda

### *Búsqueda $A^*$*

*Encuentra la salida del laberinto con el menor coste*

Belén Melián Batista

J. Marcos Moreno Vega

---

**OBJETIVO:**

Proponer, implementar y evaluar búsquedas A\* para encontrar el camino de menor coste desde el punto de entrada hasta el punto de salida de un laberinto.

**TAREAS:**

Además de las tareas descritas en el presente documento, los estudiantes tendrán que realizar las modificaciones que se planteen durante la corrección de la práctica.

**CORRECCIÓN:**

Semana del 14 al 18 de octubre.

**EVALUACIÓN:**

Código fuente y memoria: hasta 3 puntos, si se realiza la modificación correctamente. Si el día de la corrección falta algún código o este es incorrecto, la práctica se calificará como No apta.

Modificación propuesta el día de la corrección y defensa oral de la misma: hasta 7 puntos.

**LENGUAJE DE PROGRAMACIÓN:**

C++, Java o Python.

---

# Problema del camino de mínimo coste entre los puntos de entrada y salida de un laberinto

Sea dado un laberinto como el que se muestra en la Figura 1. Los puntos de entrada y salida del laberinto están representados con las letras  $S$  y  $E$ , respectivamente. Se permite moverse a través del laberinto en dirección vertical, horizontal y diagonal, lo que incurre en un coste de 5, 5, y 7, respectivamente. El objetivo del problema a resolver es encontrar el camino de mínimo coste que va desde la casilla  $S$  hasta la casilla  $E$ . Para ello, se diseñará e implementará una Búsqueda  $A^*$ . Dada una casilla  $n$ , se define la función  $f(n)$  de la siguiente manera:

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

donde  $g(n)$  es el coste acumulado de movimiento desde el punto de entrada,  $S$ , hasta la casilla  $n$ , y  $h(n)$  es una heurística para este problema. Para su definición se hará uso de la distancia de Manhattan. Por lo tanto, la función heurística para la celda  $n$ , con coordenadas en  $(x_n, y_n)$ , donde  $(x_E, y_E)$  corresponde a la ubicación de la meta,  $E$ , y  $W$  es el coste del movimiento vertical/horizontal, se define de la siguiente forma:

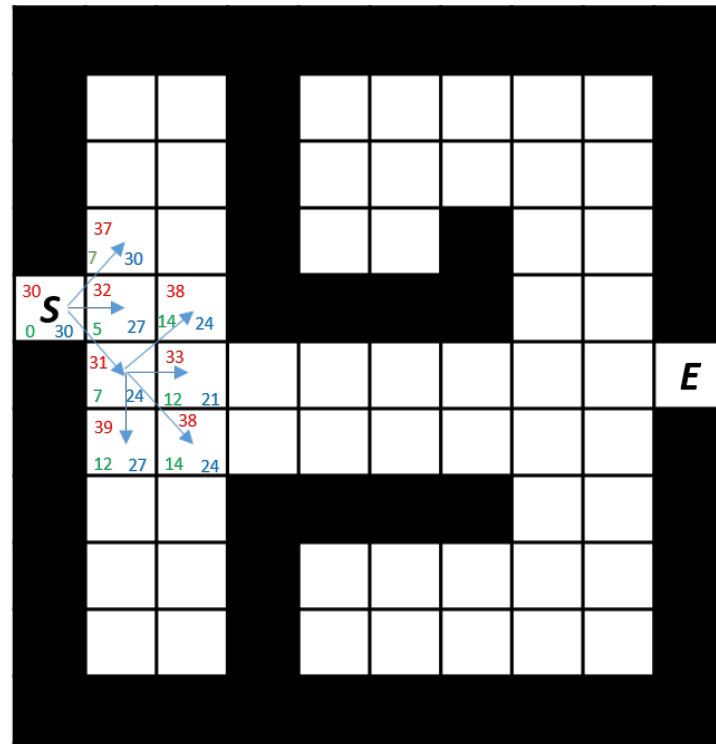
$$h(n) = (|x_E - x_n| + |y_E - y_n|) * W, W = 3.$$

La Figura 1 muestra la evaluación de las casillas vecinas al punto de entrada,  $S$ . Los valores representados en color verde muestran los resultados de la función de evaluación del movimiento,  $g(n)$ , mientras que los representados en color azul muestran los valores de la función heurística admisible,  $h(n)$ . Finalmente, los valores en color rojo representan la función de evaluación  $f(n) = g(n) + h(n)$ .

## Algoritmo $A^*$

Para realizar la implementación del algoritmo  $A^*$ , se deben seguir los siguientes pasos:

1. Calcular  $f(n)$ ,  $g(n)$  y  $h(n)$  para el punto de entrada al laberinto,  $S$ , que se inserta en la lista de nodos abiertos  $\mathcal{A}$ .
2. Repetir mientras  $\mathcal{A}$  no esté vacía.
  - (a) Seleccionar el nodo de menor coste  $f(n)$ , e insertarlo en la lista de nodos cerrados  $\mathcal{C}$ .
  - (b) Para cada nodo vecino, realizar las siguientes acciones:
    - i. Si el nodo no está ni en  $\mathcal{A}$ , ni en  $\mathcal{C}$ , su nodo padre será el nodo analizado y será insertado en  $\mathcal{A}$ . Realizar las acciones que corresponda.



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

Figura 1: Laberinto y cálculo de costes hacia los vecinos

- ii. Si el nodo está en  $\mathcal{A}$ , tener en cuenta que quizás sea necesario actualizar su coste  $g(n)$  y, por lo tanto, su padre en el camino. Nótese que se deben recalcular los costes necesarios.
3. Si  $\mathcal{A} = \emptyset$  y no se ha llegado a la salida del laberinto,  $E$ , no existe camino y se deberá mostrar en pantalla un mensaje que así lo indique.

Se deberá hacer uso de la estructura de datos adecuada para almacenar el camino óptimo y el coste del mismo. También deben ser impresos en pantalla.

## Implementación

Las instancias del problema se suministrarán en un fichero de texto con el formato mostrado en la Figura 2. El punto 3 representa el punto de entrada al laberinto, el punto 4 representa la salida, el número 1 indica que hay un obstáculo y la casilla no es transitable y, por último, el número 0 representa las casillas de paso.

1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	1	0	0	1
<b>3</b>	0	0	1	1	1	1	0	0	1
1	0	0	0	0	0	0	0	0	<b>4</b>
1	0	0	0	0	0	0	0	0	1
1	0	0	1	1	1	1	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Figura 2: Ejemplo de instancia

## Tareas

- a) Diseñar e implementar una búsqueda A\* para el problema del camino de mínimo coste entre las casillas de entrada y salida de un laberinto dado haciendo uso de la distancia de Manhattan.

Además, debe poder indicarse por teclado cuáles son las casillas de entrada y salida del laberinto.

- b) Volcar en un **fichero la salida** indicada por la **Tabla 3**, así como el **camino obtenido marcado con un \*** sobre el laberinto de entrada con el formato indicado en la **Figura 1**.
- c) Analizar el comportamiento de la búsqueda A\* considerando funciones heurísticas alternativas a la propuesta en este guión de prácticas.

## Qué debe presentar el alumno

- a) Código fuente, debidamente comentado, y fichero ejecutable.
- b) Una memoria en formato pdf en la que se describan brevemente la búsqueda

A\* diseñada enumerando las estructuras de datos usadas y cualquier elemento necesario para comprender el diseño propuesto.

- c) La memoria debe incluir también tablas o gráficas de resultados que muestren el comportamiento de la búsqueda sobre diferentes instancias del problema. En la Figura 3 se muestra el formato de estas tablas de resultados. Además, se debe **dibujar el camino obtenido marcado con un \* sobre el laberinto de entrada con el formato indicado en la Figura 1.**

Se han considerado tres laberintos (instancias  $M_1, M_2, M_3$ ), de diferentes tamaños, con  $n$  el número de filas y  $m$  el número de columnas, con varias combinaciones de casillas  $S$  y  $E$ . En la tabla se mostrará el camino de coste mínimo mínimo para ir de  $S$  a  $E$ , su coste y los nodos generados e inspeccionados por la búsqueda A\* que usa la función heurística  $h(\cdot)$ .

Búsqueda A*. Función heurística $h(\cdot)$									
<i>Instancia</i>	<i>n</i>	<i>m</i>	<i>S</i>	<i>E</i>	<i>Camino</i>	<i>Coste</i>	<i>Nodos generados</i>	<i>Nodos inspeccionados</i>	
$M_1$									
$M_1$									
$M_2$									
$M_2$									
$M_3$									
$M_3$									

Figura 3: Búsqueda A\*. Tabla de resultados