

# INTELIGENCIA ARTIFICIAL

## Algoritmos de búsqueda

### *Búsquedas no informadas*

Belén Melián Batista

J. Marcos Moreno-Vega

---

**OBJETIVO:**

Implementar búsquedas no informadas para encontrar un camino entre dos vértices de un grafo.

**TAREAS:**

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

**EVALUACIÓN:**

Correcta realización de las modificaciones propuesta el día de la corrección.

**LENGUAJE DE PROGRAMACIÓN:**

C++, Java o Python

---

## Problema del camino entre dos vértices de un grafo

Sea dado un grafo  $G = (V, E)$ , donde  $V$  es el conjunto de vértices y  $E$  es el conjunto de aristas ( $|V| = n, |E| = m$ ). Cada arista  $(i, j) \in E$  tiene asociada una distancia o coste  $d(i, j)$ . Se desea encontrar un camino que conecte el vértice origen  $v_0$  con el vértice destino  $v_d$ .

## Implementación

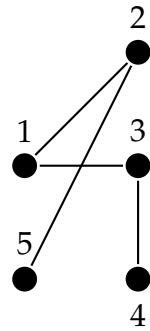
Las instancias del problema se suministrarán en un fichero de texto con el siguiente formato: en la primera fila se encuentra el número de vértices,  $n$ ; a continuación, se enumeran las distancias,  $d(i, j)$ , entre los pares de vértices. Se asume que las distancias son simétricas, es decir, que  $d(i, j) = d(j, i)$ ,  $\forall i, j \in V$ . Además,  $d(i, i) = 0$ ,  $\forall i \in V$  y  $d(i, j) = -1$  si no hay una arista que conecte al vértice  $i$  con el vértice  $j$ .

Por ejemplo, si  $n = 5$ , el fichero de texto para el grafo de la figura 1(a) contendría los datos mostrados en la figura 1(b) (solo la primera columna; la segunda describe qué representa cada dato):

## Tareas

- Implementar una búsqueda en amplitud y una búsqueda en profundidad para el problema de encontrar un camino que conecte dos vértices de un grafo.

Debe poder indicarse, cómodamente, cuáles son los vértices origen y destino.



(a) Grafo

5	<i>/*</i> número de vértices
1.225	<i>/*</i> $d(1,2) = d(2,1)$
1.000	<i>/*</i> $d(1,3) = d(3,1)$
-1	<i>/*</i> $d(1,4) = d(4,1)$
-1	<i>/*</i> $d(1,5) = d(5,1)$
-1	<i>/*</i> $d(2,3) = d(3,2)$
-1	<i>/*</i> $d(2,4) = d(4,2)$
2.236	<i>/*</i> $d(2,5) = d(5,2)$
1.000	<i>/*</i> $d(3,4) = d(4,3)$
-1	<i>/*</i> $d(3,5) = d(5,3)$
-1	<i>/*</i> $d(4,5) = d(5,4)$

(b) Formato del fichero de datos

Figura 1: Grafo y su representación

El programa suministrará el camino encontrado, el coste de dicho camino y los conjuntos de nodos generados y analizados durante el proceso de búsqueda. La salida se derivará a un fichero que almacene toda esta información. En la tabla 1 se muestra un ejemplo de salida.

Número de nodos del grafo: 5
Número de aristas del grafo: 4
Vértice origen: 1
Vértice destino: 4
Iteración 1
Nodos generados: 1
Nodos inspeccionados: -
Iteración 2
Nodos generados: 1, 2, 3
Nodos inspeccionados: 1
Iteración 3
Nodos generados: 1, 2, 3, 5
Nodos inspeccionados: 1, 2
...
...
...
Camino: 1 - 3 - 4
Costo: 2.00

Tabla 1: Fichero de resultados