

Algoritmos de Búsqueda Ciega

Stalin Muñoz Gutiérrez

Centro de Ciencias de la Complejidad
Universidad Nacional Autónoma de México (UNAM)

Hoy trataremos sobre algoritmos de búsqueda ciega, también denominados de búsqueda no informada.

Estos algoritmos exploran el espacio de estados hasta encontrar una solución o consumir los recursos computacionales establecidos o disponibles para realizar la tarea.

El nombre viene del hecho de que no utilizan información para guiar la búsqueda hacia el objetivo.

La búsqueda se detiene cuando el objetivo se logra, por ello debe especificarse una función de paro la cual distingue a los estados meta de los estados intermedios.

Los algoritmos de búsqueda ciega deciden sobre la expansión de estados basados únicamente en el conocimiento de los nodos descubiertos durante la exploración y en su caso, en el costo acumulado sobre las trayectorias desde el estado inicial.

Definición de problema



Algoritmos de Búsqueda Ciega

2018-09-13

Definición de problema

Definición de problema

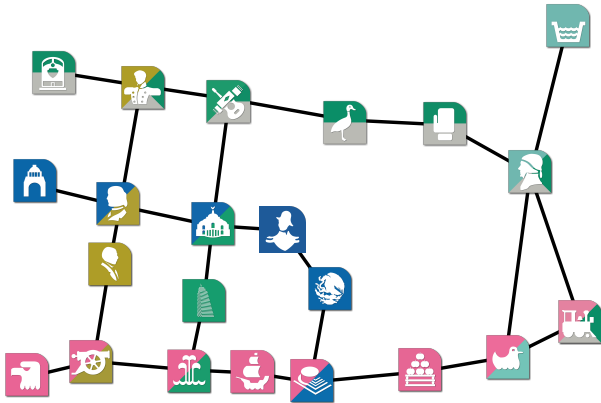


Para poder aplicar los algoritmos que veremos en este curso tenemos que definir que es un problema de manera concreta.

Para ilustrarlo usaremos el ejemplo del agente inteligente que resuelve rutas en el Sistema de Transporte Colectivo metro de la Ciudad de México.

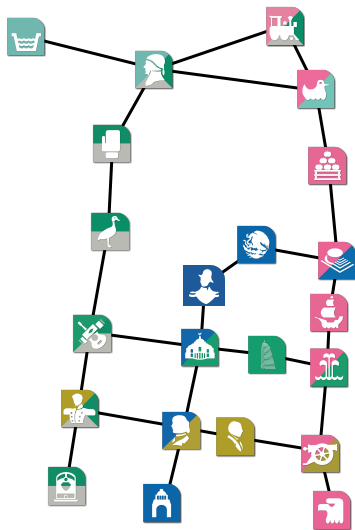
Lo vamos a ilustrar con un número de estaciones reducido.

- Definición de problema



Vamos a abstraer el problema identificando el grafo de estados-acciones.

Definición de problema



Definición de problema

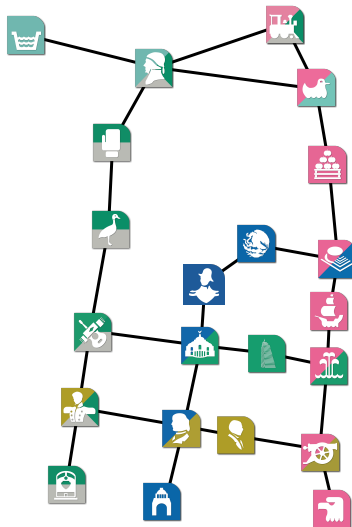


Rotaremos el mapa 90 grados a la izquierda para facilitar su lectura.



- Definición de problema

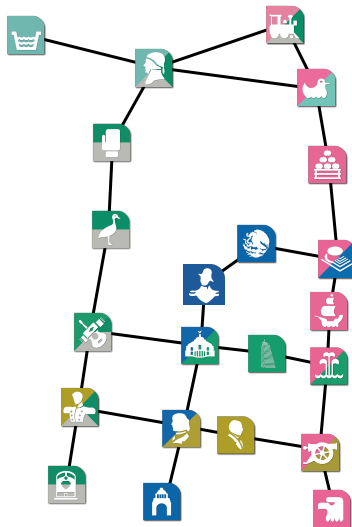
Formalmente el problema quedará determinado por los siguientes elementos.





Problema

- Espacio de estados S



El espacio de estados S .

Tenemos que pensar que este espacio sea numerable. Idealmente suficientemente pequeño para que los recursos computacionales disponibles permitan la búsqueda de soluciones.

Definición de problema

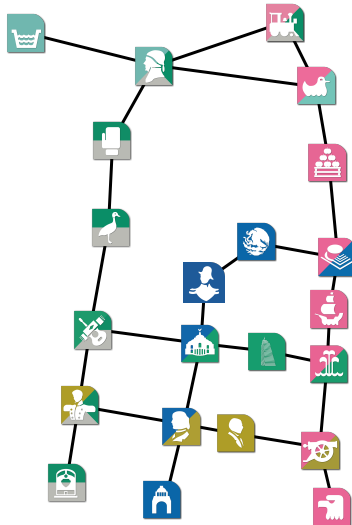
Problema

- Espacio de estados S
 $S = \{ \text{cat}, \text{perro}, \text{humano}, \dots \}$



En nuestro mapa, los estados representan las estaciones del metro.

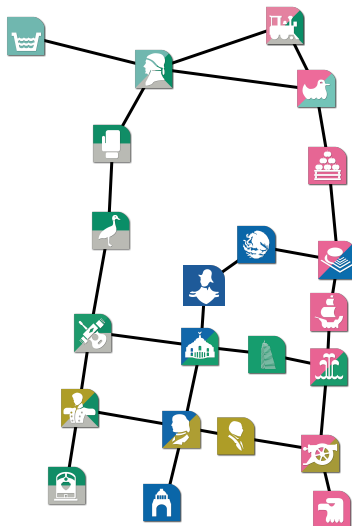
- $$S = \{ \text{Icon 1}, \text{Icon 2}, \text{Icon 3}, \dots \}$$



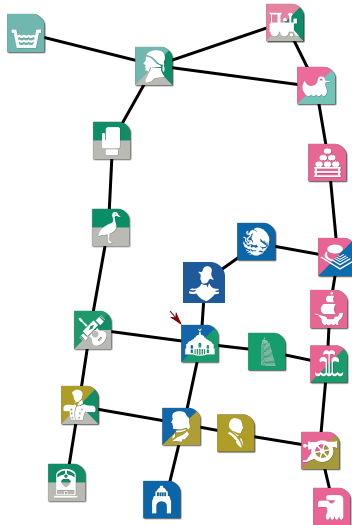


Problema

- Espacio de estados S
- Estado inicial $s_0 \in S$



Debemos conocer el estado inicial en el que se encuentra el agente.
De no conocerse se requiere de otro tipo de representaciones y algoritmos.

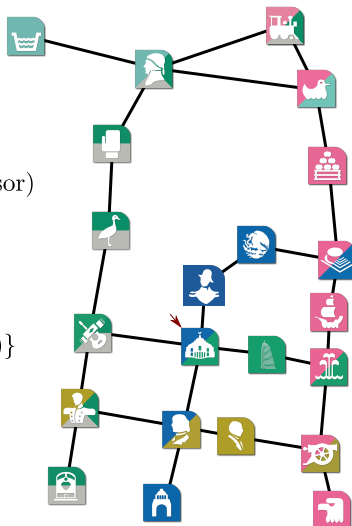

$$S_0 = \text{[Image of a building with a dome and minaret, likely a mosque or university building.]}$$


En nuestro ejemplo el estado inicial será la estación Bellas Artes.

- Definición de problema

Por brevedad se ilustra la expansión, omitiendo la palabra ir y poniendo el estado al que llegamos cuyo subíndice es el estado del que venimos.

- $$\begin{aligned} & \text{expandir} : S \rightarrow A \text{ donde} \\ & A = \{(a, s) \mid a \in A \wedge s \in S\} \\ & \text{expandir}(\text{person}) = \{ \\ & (\text{ir}(\text{person}, \text{sea}), \text{sea}), (\text{ir}(\text{person}, \text{templo}), \text{templo}) \} \\ & \text{expandir} = \{ \text{sea}, \text{templo} \} \end{aligned}$$

2018-09-13⁶

Definición de problema

Problem

- Espacio de estados S
- Estado inicial $s_0 \in S$
- Acciones del agente (Función sucesor)
 $expandir : S \rightarrow A$ donde
 $A = \{(a, s) \mid a \in A \wedge s \in S\}$
 $expandir = \{$
 $(a_1, s_1), (a_2, s_2), (a_3, s_3), (a_4, s_4), (a_5, s_5), (a_6, s_6), (a_7, s_7), (a_8, s_8), (a_9, s_9), (a_{10}, s_{10})\}$
 $expandir = \{$
 $(a_1, s_1), (a_2, s_2), (a_3, s_3), (a_4, s_4), (a_5, s_5), (a_6, s_6), (a_7, s_7), (a_8, s_8), (a_9, s_9), (a_{10}, s_{10})\}$



└ Definición de problema

Definición de problema

- Problema
 - Estado inicial $s_0 \in S$
 - Acciones del agente (Función sucesor)
 - $expandir : S \rightarrow A$ donde
 - $A = \{ \langle a, a \rangle \mid a \in A \wedge a \in S \}$
 - $expandir(\langle a, a \rangle) = \{ \langle a_1, a_1 \rangle, \dots, \langle a_n, a_n \rangle \}$
 - Función meta (o de paro):
 - $parar : S \rightarrow \{cierto, falso\}$
 - $parar(x) = \begin{cases} \text{cierto,} & \text{if } s = \text{estado meta} \\ \text{falso,} & \text{otro caso} \end{cases}$

[illegible]

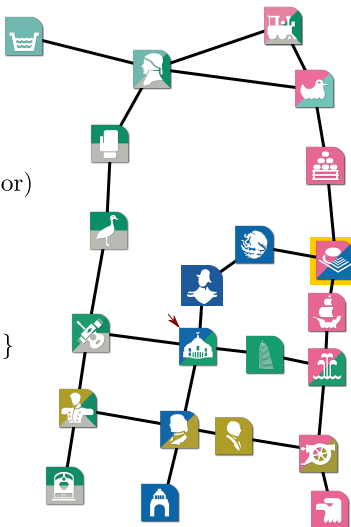
The diagram shows a complex network of interconnected nodes. Each node contains a small icon representing a specific entity or service, such as a person, a building, a car, or a document. The nodes are arranged in a non-linear fashion, with multiple paths connecting them. Some nodes are highlighted with red borders, indicating they might be key components or entry points in the network.

[illegible]

-
- The diagram shows a complex network of interconnected nodes. Each node contains a small icon representing a specific entity type, such as a person, a building, a car, or a document. The nodes are arranged in a non-linear fashion, with multiple paths connecting them. Some nodes are highlighted with thicker borders or larger sizes, indicating their importance or central role in the network.

[illegible]

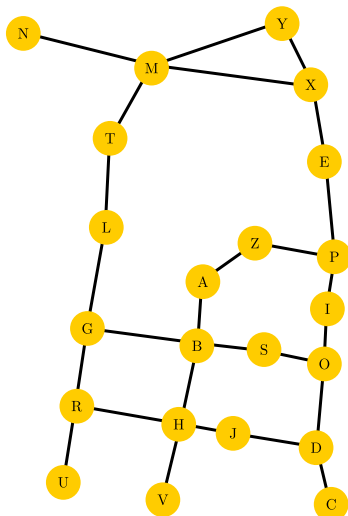
The diagram shows a complex network of interconnected nodes. Each node contains a small icon representing a specific entity type, such as a person, a building, a car, or a document. The nodes are arranged in a non-linear fashion, with multiple paths connecting them. Some nodes are highlighted with thicker borders or larger sizes, indicating their importance or central role in the network.



Resolver el problema, en este caso, es encontrar la secuencia de acciones que consigue esta meta.

Estructuras de datos

Encontrar la ruta de B a P



En nuestro ejemplo nos interesa encontrar la ruta entre las estaciones Bellas Artes y Pino Suarez.

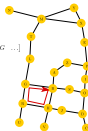
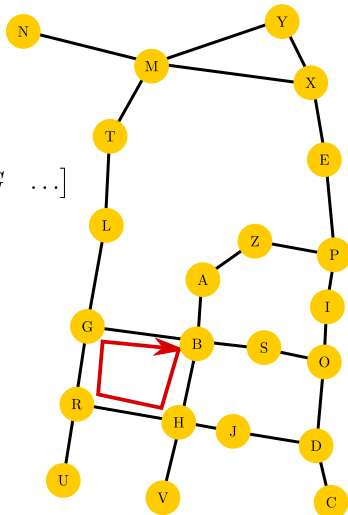
Solo usaremos letras para identificar los estados.

Estructuras de datos

Encontrar la ruta de B a P

- Necesitamos evitar caer en ciclos infinitos.

$[B \ H \ R \ G \ B \ H \ R \ G \ \dots]$



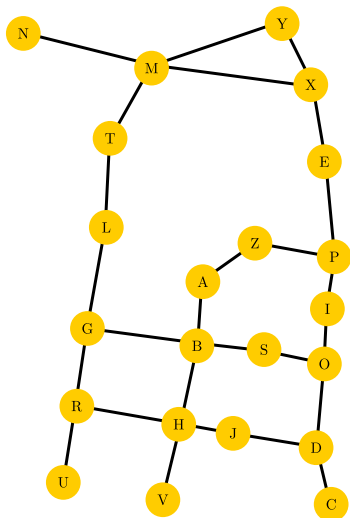
Utilizando la función *expandir* los algoritmos van a explorar el espacio de estados.

Hay que tener cuidado en no caer en ciclos infinitos.

Estructuras de datos

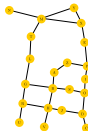
Encontrar la ruta de B a P

- Necesitamos evitar caer en ciclos infinitos.
- Esto significa poder recordar los estados anteriores.



Estructuras de datos

- Encontrar la ruta de B a P
- Necesitamos evitar caer en ciclos infinitos.
 - Esto significa poder recordar los estados anteriores.



Para ello el algoritmo tendrá que recordar los estados que ya visitó.
Esto significa que tiene que usar una memoria.

└ Estructuras de datos

Los algoritmos que veremos, utilizarán diferentes estructuras de datos para almacenar los estados.

- Como veremos en su momento. Esto tendrá una repercusión importante en su comportamiento y en los recursos de computo que requiere para resolver el problema.

Dependiendo del algoritmo se usarán pilas, colas, colas de prioridad, tablas de dispersión etc.

