

Algoritmos de Búsqueda

Un algoritmo de búsqueda es aquel que está diseñado para localizar un elemento con ciertas propiedades dentro de una estructura de datos; por ejemplo, ubicar el registro correspondiente a cierta persona en una base de datos, o el mejor movimiento en una partida de ajedrez.

La variante más simple del problema es la búsqueda de un número en un vector.

Un problema típico de la Inteligencia Artificial consiste en buscar un estado concreto entre un conjunto determinado, al que se le llama espacio de estados. Imaginemos, por ejemplo, una habitación con baldosines en la que hay un libro. Un robot se desea desplazar por la habitación con el fin de llegar a dicho libro. ¿De qué manera lo hará? En este punto es donde entran en juego las estrategias y los algoritmos de búsqueda.

Cuando el sistema agente (en este caso, el robot) posee algún tipo de información del medio, se utilizan técnicas de búsquedas informadas; sin embargo, si carece de conocimiento alguno, se deberán emplear algoritmos de búsqueda no informadas. En nuestro ejemplo, y para este último caso, podemos imaginar un robot que no posea ningún tipo de visión artificial, que únicamente sea capaz de moverse en horizontal o vertical de un baldosín a otro y detectar si en el baldosín se halla el libro.

Se utiliza cuando el vector no está ordenado o no puede ser ordenado previamente. Consiste en buscar el elemento comparándolo secuencialmente (de ahí su nombre) con cada elemento del vector hasta encontrarlo, o hasta que se llegue al final. La existencia se puede asegurar cuando el elemento es localizado, pero no podemos asegurar la no existencia hasta no haber analizado todos los elementos del vector. A continuación se muestra el pseudocódigo del algoritmo:

Datos de entrada:

```
vec: vector en el que se desea buscar el dato
tam: tamaño del vector. Los subíndices válidos van desde 0 hasta
tam-1 inclusive. Puede representarse así: vec[0...tam) o
vec[0...tam-1].
dato: elemento que se quiere buscar.
```

Variables

```
pos: posición actual en el vector
pos = 0
while pos < tam:
    if vec[pos] == dato:
        Retorne verdadero y/o pos,
    else:
        pos = pos + 1
Fin (while)
Retorne falso,
```