



UNIVERSIDAD AUTONOMA DE AGUASCALIENTES

CENTRO DE CIENCIAS BASICAS

INGENIERIA EN COMPUTACION INTELIGENTE

3°A

**“RESUMEN A12 METODOS DE BUSQUEDA”**

SANDOVAL PEREZ JOSE LUIS 261731

DR.ALEJANDRO PADILLA DIAZ

## **Búsqueda en profundidad(DFS)**

Este método de búsqueda utilizado para búsqueda en árboles. Inicialmente se expande el nodo inicial y se generan sus sucesores, después, se expande el nodo más recién generado, si el nodo no tiene sucesores se expande a otro nodo diferente.

## **Backtracking simple**

Es un método de búsqueda en profundidad que termina encontrando la primera solución ya sea con el costo optimo o no. No utiliza información heurística.

## **Búsqueda en profundidad Branch-and-Bound**

Este método continua después de encontrar una solución, de manera que constantemente va actualizando el mejor camino descartando así los caminos soluciones que conducen a una peor solución. Es un algoritmo dependiente de la función heurística.

## **Iterative Deepening A\*(IDA\*)**

Se garantiza solución si cada nodo tiene número de descendiente finito. Utiliza 3 parámetros que formulan la solución:

$$l(x) = g(x) + h(x)$$

En donde; g es una estimación de lo que se ha recorrido, h es una función heurística que representa una estimación de lo que queda por recorrer y finalmente l es una estimación del camino solución desde el nodo inicial pasando por x.

## **Búsqueda primero en anchura**

Es un algoritmo para recorrer o buscar elementos en un grafo. Este algoritmo comienza desde la raíz y después se exploran todos los vecinos de este nodo. Después de esto se selecciona otro nodo y se repite el proceso, así, hasta terminar de recorrer todo el árbol.

Su implementación se hace presente en búsqueda en árboles con una estructura FIFO.

Una evaluación con este algoritmo se divide en 2;

Completa: Si el nodo mas superficial está en cierta profundidad finita se lo encontrara luego de expandir los nodos más superficiales.

Optima: Es optimo si el coste del camino es una función no decreciente de la profundidad del nodo.

Es importante preocuparse por la memoria ya que este algoritmo es exponencial.