

Ambos tipos de búsqueda son aplicables tanto para grafos dirigidos y no dirigidos.

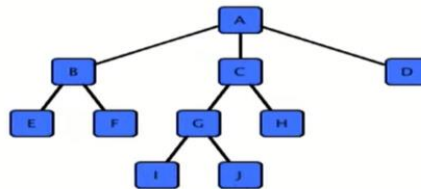
Búsqueda en Profundidad

Una búsqueda en profundidad (DFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo). Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

La búsqueda en profundidad se usa cuando queremos probar si una solución entre varias posibles cumple con ciertos requisitos; como sucede en el problema del camino que debe recorrer un caballo en un tablero de ajedrez para pasar por las 64 casillas del tablero.

Algoritmo DFS

Desde cada vértice consideraremos el primero de los vértices que dependan de él.



A, B, E, F, C, G, I, J, H, D.

2.6. Algoritmo DFS (Depth First Search)

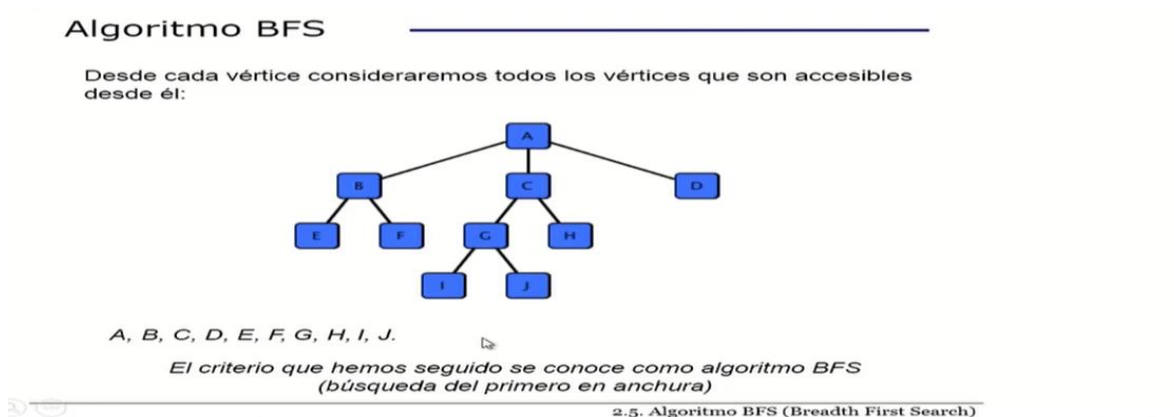
Aplicaciones de DFS

En un grafo acíclico dirigido, es decir un conjunto de nodos donde cada nodo tiene una sola dirección que no es consigo mismo, se puede realizar un ordenamiento topológico mediante el algoritmo DFS. Por ejemplo, se puede aplicar para organizar actividades que tienen por lo menos alguna dependencia entre sí, a fin de organizar eficientemente la ejecución de una lista de actividades.

Búsqueda en Anchura

Una búsqueda en anchura (BFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo, comenzando en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo), para luego explorar todos los vecinos de este nodo. A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el grafo. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

La búsqueda por anchura se usa para aquellos algoritmos en donde resulta crítico elegir el mejor camino posible en cada momento del recorrido.



Aplicaciones de BFS

En grafos no ponderados (los nodos no tienen una medida o peso), mediante el algoritmo BFS es posible encontrar el camino más corto. Por ejemplo, si tenemos un conjunto de actividades representadas por nodos y deseamos llegar a una actividad específica realizando la menor cantidad de actividades posibles, podemos usar este algoritmo; en el caso que tengamos un grafo ponderado para el ejemplo anterior donde las actividades tengan un tiempo estimado, podemos modificar el algoritmo BFS, llegando a obtener el algoritmo Dijkstra.

Conclusión

A través de los algoritmos mencionados en este artículo, se puede encontrar caminos o recorridos entre dos o más nodos dependiendo de la naturaleza del problema. Por ejemplo, se puede usar el algoritmo BFS en mapas de Google para encontrar rutas óptimas entre puntos turísticos de interés o para hacer búsquedas de textos en una página web por medio de un Web Crawler; por medio del algoritmo DFS, se puede resolver o simular juegos como laberintos o ajedrez.