

PROTOCOLO

Comparativa de la eficiencia en tiempo de los algoritmos BFS y DFS

Sergio Sanz Martínez, Hugo Pelayo Aseko, Isha Shafqat
Noviembre 2021

Objetivos

Cuando nos encontramos programando alguna tarea, encontrar la solución más rápida es una de las prioridades. Para ello es importante tener conocimientos de diferentes algoritmos específicos y de esta manera saber cual es el mejor para cada problema.

En este trabajo nos encontraremos trabajando con grafos. Los grafos permiten representar conexiones entre diferentes nodos, un ejemplo sería la conexión entre ordenadores por internet o las diferentes ciudades de un país.

El objetivo principal de este trabajo es comparar dos algoritmos populares que se usan para recorrer grafos. Estos son el algoritmo de búsqueda en profundidad o DFS (Depth First Search) y el algoritmo de búsqueda en anchura o BFS (Branch First Search).

Lo que compararemos en concreto será el tiempo que tardan los dos algoritmos en, partiendo de un nodo ***K***, encontrar un nodo ***P***.

Variables recogidas

Nuestra variable de interés ***X*** será el tiempo de ejecución del algoritmo en segundos. La variable de interés ***Y*** será el tipo de algoritmo utilizado para la búsqueda en grafos (DFS o BFS). Por otra parte incluimos una variable ***Z*** que representa el orden de los grafos (número de nodos que contienen).

Recogida de datos

Hemos generado al azar 40 grafos de hasta 150 nodos y con un máximo de $(n \cdot (n - 1)) / 4$ aristas, donde n es el número de nodos.

Para la implementación de los algoritmos utilizaremos el lenguaje de programación de alto nivel C++. C++ es un lenguaje que soporta la programación orientada a objetos (OOP) y que nos servirá de herramienta para representar internamente los grafos mediante listas de adyacencias y posteriormente la implementación de los algoritmos de búsqueda BFS y DFS.

Un grafo es una pareja de conjuntos $G = (V, A)$, donde V es el conjunto de vértices y A es el conjunto de aristas, este último es un conjunto de pares de vértices de la forma (u, v) tal que $u, v \in V$, estos pares representan la conexión entre dos vértices.

Los grafos son estructuras que se pueden representar de formas variadas, en este estudio en concreto consideramos la representación mediante listas de adyacencia.

Nuestro programa se ejecutará sobre un computador con las siguientes características: Intel Core i5 7300HQ a 2.5Ghz, 8GB de memoria RAM y sistema operativo Microsoft Windows 10 PRO.

Recogeremos el orden y medida de cada uno de nuestros grafos y utilizaremos BFS y DFS sobre cada uno. A continuación mediremos el tiempo que tardan nuestras implementaciones en, partiendo de un mismo nodo ***K***, encontrar un nodo ***P***.

La información sobre los algoritmos ha sido obtenida de los siguientes enlaces:

BFS: https://en.wikipedia.org/wiki/Breadth-first_search

DFS: https://en.wikipedia.org/wiki/Depth-first_search

Muestra

Para conseguir una muestra aleatoria de la población de grafos que vamos a tratar, hemos decidido obtener grafos con diferente número de vértices cada uno y generar el orden y las aristas de forma aleatoria. De esta manera, tanto el orden como los propios grafos están generados de forma aleatoria proporcionando una muestra de 40 grafos de diferente orden.