

Algoritmos y Estructuras de Datos

Proyecto Final: DFS.

Marco Antonio Heredia Velasco

Fecha de entrega: Miércoles 9 de julio (se revisará durante la clase).

1. Instrucciones de entrega

Las instrucciones de entrega son las mismas que con las prácticas: Se deberá realizar en equipos de a lo más 3 personas. Todos los archivos deben estar contenidos en un directorio cuyo nombre estará formado por el primer apellido de los integrantes del equipo. Dentro debe haber 2 subdirectorios: **docs** y **fuentes**. Se debe empaquetar esta estructura de directorios en un archivo comprimido, llamado igual que el directorio principal, pero con extensión determinada por el compresor utilizado (Ej. **HerediaUrrutiaPerez.zip**). Aunque el proyecto se revisará en clase, deben enviarme dicho archivo comprimido antes de que ésta inicie, a la dirección **hvma@correo.azc.uam.mx** con asunto: *Proyecto Final AED*.

El proyecto debe ser elaborado en C, C++ o Java. Los proyectos que no cumplan con estas instrucciones, no serán revisados.

2. Programa

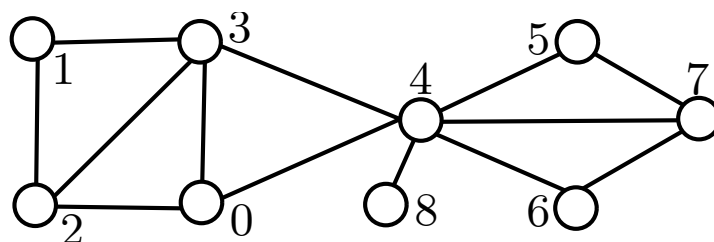
Como se vió en clase, el algoritmo de Búsqueda en profundidad (DFS) es bien conocido y ampliamente utilizado para recorrer los vértices de una gráfica en forma ordenada.

La estrategia que sigue DFS es ir recorriendo un camino predeterminado en la gráfica hasta que ya no pueda seguir ampliándolo, cada vez que se da esta situación, regresa sobre el mismo camino hasta encontrar un vértice que le permita llegar a vértices que no han sido visitados. Si se deja correr DFS sobre una gráfica conexa, el resultado será un árbol generador (que contiene todos los vértices), con raíz en el vértice en que se inició el proceso.

Aunque hay diversas versiones del algoritmo, hay una en particular que se ayuda de una pila para mantener el control de los vértices por visitar (vista en clase).

2.1. Actividades

El objetivo de este proyecto es hacer un programa llamado *DFS* que corra dicho algoritmo en la siguiente gráfica G , empezando desde el vértice 0:



El resultado que entregará el programa será el árbol DFS que se obtiene al correr dicho algoritmo en G .

2.2. Especificaciones

El código fuente de su programa deberá estar en el directorio **fuentes** y debe estar ampliamente comentado. Sólo deben mandar los archivos con el código fuente, no debe incluir ningún ejecutable.

Se debe utilizar la versión de DFS vista en clase, que hace uso de una pila.

El programa deberá estar constituido de al menos 2 archivos, uno que contenga la implementación de una pila de enteros (por ejemplo **pila.c**) y otro que contenga tanto la implementación de DFS como la representación de la gráfica de arriba (por ejemplo **dfs.c**). Si se desea se puede implementar la estructura de datos de la gráfica en un archivo por separado (por ejemplo **grafica.c**). Desde el archivo que contenga el main, se debe importar (**#include**) los demás archivos implementados para usarlos desde ahí.

Para representar la gráfica se debe hacer uso de una de las 4 formas vistas en clase: Matriz o lista de adyacencias, ó Matriz o lista de incidencias. A la hora de implementar alguna de estas representaciones, pueden suponer que el número de vértices de la gráfica siempre es 9.

El algoritmo DFS visto en clase guarda el árbol resultante en un arreglo en que se especifica quién es el padre de cada vértice. El programa debe dar como resultado el contenido de este arreglo, diciendo cosas como: “El padre del vértice 1 es el vértice 5.” ó “El vértice 0 es la raíz y no tiene padre.”.

3. Documentos

El directorio `docs` deberá contener un archivo `README.txt` en texto plano, indicando el nombre completo de los integrantes del equipo, el lenguaje de programación utilizado, el sistema operativo en se probó el programa, y la forma adecuada de compilar y correr su programa. También se debe incluir una explicación de la forma en que fue implementado el programa (qué funciones nuevas hicieron, cómo funcionan, qué modificaciones hicieron a su pila, etc.).