



Proyecto

Objetivo General

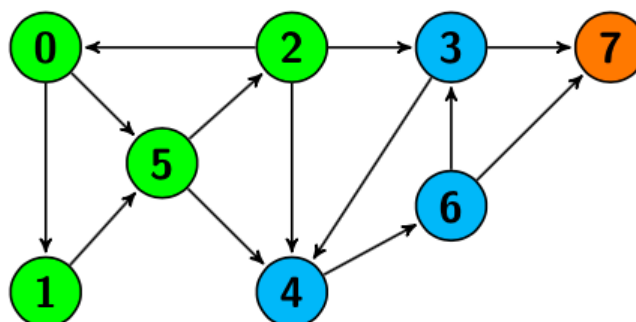
Aplicar los conceptos generales sobre la teoría de grafos enfocados a programación.

Objetivos Específicos

- Comprender la utilidad de la teoría de grafos.
- Aplicar la teoría de grafos empleando distintos lenguajes de programación.
- Demostrar por medio de un entorno gráfico las aplicaciones de la teoría de grafos.

Teoría de Grafos

La teoría de grafos juega un papel importante en la fundamentación matemática de las Ciencias de la Computación. Los grafos constituyen una herramienta básica para modelar fenómenos discretos y son fundamentales para la comprensión de las estructuras de datos. Los grafos no son más que colecciones de objetos llamados “vértices” conectados por líneas llamadas “aristas” que pueden tener una orientación (dirección asignada), si las aristas poseen dirección se dice que es un grafo dirigido, y si no poseen dirección es un grafo no dirigido.



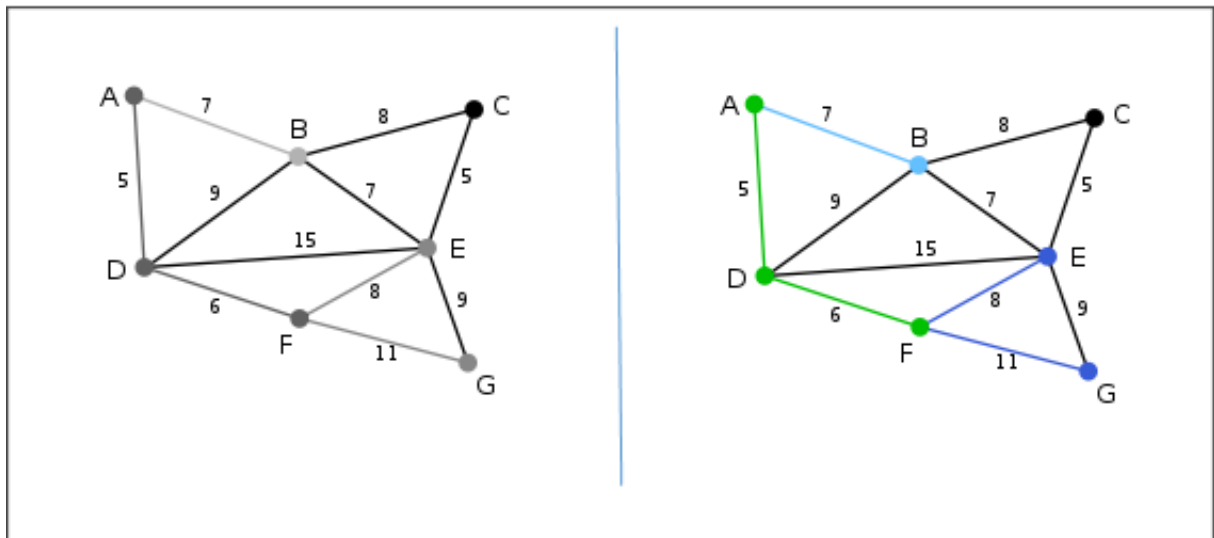
La teoría de grafos tiene varios algoritmos, que se utilizan para resolver situaciones, encontrar sub-grafos o caminos cortos. Entre los algoritmos más utilizados se encuentran:

- ☐ Algoritmo de búsqueda en anchura
- ☐ Algoritmo de búsqueda en profundidad
- ☐ Árbol generador minimal
- ☐ Algoritmo de Prim
- ☐ Algoritmo de Kruskal

Una parte importante en la comprensión de los distintos algoritmos es la visualización de la aplicación de este mismo. Por esto, se le solicita que realice un programa que ejemplifica de forma gráfica el funcionamiento del mismo.

A cada grupo de trabajo se le asignará un algoritmo de los listados anteriormente y respecto a este debe de realizar un programa informático con una interfaz gráfica que permita ingresar un grafo, y visualizar la aplicación y funcionamiento del mismo. La forma en que se ingresan los vértices y aristas puede realizarlo como guste y más se ajuste al tipo de algoritmo solicitado. De igual manera, deberá desplegar el grafo original ingresado, y el grafo con el algoritmo aplicado, en la manera que considere que sea apreciable.

El siguiente ejemplo muestra una interfaz gráfica con la aplicación del algoritmo de Prim.



<p>Vértice</p> <input type="text"/>		<p>Agregar</p>		<p>Vértices</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div>		<p>Aristas</p> <div style="border: 1px solid black; height: 100px; width: 100%;"></div>		<p>Generar</p>	
<p>Arista</p> <input type="text"/>		<p>Peso</p> <input type="text"/>							
<p>Ej. A->B</p>		<p>Agregar</p>							

En este mockup, se solicita el vértice que va ingresar al programa. También solicita la Arista en el formato A->B y un espacio para el peso de la arista. En el lado derecho se tiene un resumen de los vértices y las aristas que se han ingresado; y finalmente un botón que genera el grafo solicitado. En la parte superior se puede apreciar ambos grafos. Del lado izquierdo el grafo ingresado y a su derecha el grafo con el algoritmo de Prim aplicado.

No todos los algoritmos son aplicables a los mismos grafos, por lo que no todos los programas requieren de los mismos campos para el ingreso del grafo. Los algoritmos de búsqueda no requieren de peso en sus aristas. Los árboles AVL no requieren peso ni ingresar aristas directamente.

Debe ajustar su programa a lo que su algoritmo requiere para funcionar correctamente.

Instrucciones:

- El lenguaje de programación deberán elegir entre:
 - Java, Python o Go.
- El programa debe de tener una interfaz gráfica; tanto para ingresar el grafo como para poder visualizar el algoritmo ejemplificado.
- Para realizar el grafo se le recomienda la utilización de la herramienta Graphviz. Si desea realizar un grafo interactivo en el programa, puede hacerlo.
- Puede apoyarse con la utilización de código existente respecto a los algoritmos.
- Realizar el proyecto parejas. Anotarse en siguiente link (La hoja de parejas) para asignar el número de pareja y el tema a realizar(El tema sera asignado por el auxiliar del curso), los que no tengan pareja porfavor apuntarse y dejar un numero de telefono para que se puedan contactar:
<https://docs.google.com/spreadsheets/d/1UQAonXSdbBmEObpo8UXsdluNw26co2r3ZnUmnDDaeUM/edit?usp=sharing>
- Deberá generar un programa funcional que evidencie el funcionamiento del algoritmo asignado.

Entregables

- Ejecutable del programa realizado.
- Explicar el funcionamiento técnico y el funcionamiento general del programa (Como un manual técnico o de usuario).

Restricciones:

- La entrega del proyecto es en parejas.
- Se debe utilizar uno de los lenguajes autorizados.
- Se debe tener un entorno gráfico para el programa, no se aceptarán programas de línea de comando.
- Debe evidenciar el algoritmo por medio de los grafos generados.
- Las copias detectadas tendrán nota de 0.

Formato de Entrega:

- Archivo rar/zip que incluya el programa ejecutable y la explicación del funcionamiento.
- Enlace del repositorio en github.

El proyecto completo se entregará por medio de uedi:

28 de abril de 2023