RELATÓRIO TRABALHO PRÁTICO DE GRAFOS

Nome: Stevan Santana dos Santos

Matrícula: 2022403BCC

Nome: Caio Rodrigues da Silva

Matrícula: 2012913BCC

Sumário

1 - Decisões de Projeto

Para este projeto de Algoritmos Em Grafo foi decidido o uso de ferramentas e bibliotecas da linguagem C, sendo toda estrutura feita em códigos e algoritmos em formatos que são mais apropriados a está linguagem, decidida pelos dois integrantes da dupla antes ao início do projeto visando completar todas as exigências e funções propostas. Serão utilizados de menus auxiliares para o uso e testes das funções, assim como facilitamento de interação com código.

2 – Biblioteca

A entrega de uma biblioteca com foco em manipulação de grafos através de suas funcionalidades é um dos objetivos do projeto, e nela conteremos todas as funções preparadas para representar e utilizar de algoritmos em grafos tanto complementares quanto as especificadas no arquivo do trabalho. É previsto as funções implementadas e desenvolvidas na biblioteca serem:

- Entrada;
- Saida;
- Representação de grafos;
- Busca em Grafos: largura e profundidade
- Componentes Conexos;
- Grafos com peso;
- Distância e caminho mínimo;

- Árvore geradora mínima (MST);
- Distância média.

3 - Implementações

Todas as implementações incluindo as que estão previstas devem ser incluídas a biblioteca, os códigos e funções realizadas no projeto seguem:

3.1 - Primeira Parte do Projeto

main ( ) - Utilizada para chamar o menu de opções.

menu ( ) - Apresenta opções para trabalhar com grafos em ler grafos, representar grafos, buscas em grafos e descobrir componentes conexos de um grafo.

buscaBinaria ( ) - Implementação de um algoritmo de busca binaria a ser usado em outras funções.

lerGrafo ( ) - Apresenta opções dos grafos a serem lidos e passa para a função escolherComSemPeso( ).

- **escolherComSemPeso ( ) -** Apresenta a opção de manipular grafos com ou sem peso, chama lerGrafoComPeso ( ) e lerGrafoSemPeso ( )
- **lerGrafoComPeso ( ) -** Lê um grafo com peso com valores associados as arestas, criando um arquivo de saída.
- **lerGrafoSemPeso ( ) -** Lê um grafo sem consideração de peso, criando um arquivo de saída.

escolherGrafoRepresentação ( ) - Apresenta opções dos grafos a serem representados e passa para a função escolherModoRepresentação( ).

- **escolherModoRepresentação ( ) -** Apresenta a opção de representar um grafo em matrizes ou listas com e sem peso, chama matrizAdjacenciaComPeso( ), matrizAdjacenciaSemPeso( ), listaAdjacenciaComPeso( ) e listaAdjacenciaSemPeso.
- **matrizAdjacenciaComPeso ( ) -** Representa um grafo por uma matriz de adjacência com peso.
- **matrizAdjacenciaSemPeso ( ) -** Representa um grafo por uma matriz de adjacência sem peso.
- **listaAdjacenciaComPeso ( ) -** Representa um grafo por uma lista de adjacência com peso.
- **listaAdjacenciaSemPeso ( ) -** Representa um grafo por uma lista de adjacência sem peso.

escolherGrafoBusca ( ) - Apresenta opções dos grafos a serem buscados e passa para a função escolherModoBusca( ).

- **escolherModoBusca ( ) -** Apresenta a opção de buscar um grafo em algoritmos de largura e profundidade, chama buscaEmLargura( ) e buscaEmProfundidade( ).

- **buscaEmLargura ( ) -** Recebe um vértice inicial pelo usuário, realizando uma busca em largura e gerando um arquivo de saída com informações impressas.
- **buscaEmProfundiade ( ) -** Recebe um vértice inicial pelo usuário, realizando uma busca em profundidade e gerando um arquivo de saída com informações impressas.

escolherGrafoComponenteConexo ( ) - Apresenta opções de grafos a serem escolhidos para achar suas componentes conexas e passa para a função grafoComponenteConexo().

- **grafoComponenteConexo() -** Recebe o arquivo grafo que o usuário escolheu para achar suas componentes conexas, número de vértices de cada componente e os vértices propriamente ditos e os imprime na tela.

3.2 - Segunda Parte do Projeto

3.2.1 - A biblioteca é capaz de representar grafos não-direcionados que possuem pesos nas arestas. Basta o usuário informar o grafo com os seus respectivos pesos e a funcionalidade que ele quer que a biblioteca faça que poderá ser ler os dados do grafo em um arquivo ou representar o grafo pela matriz de adjacência com peso ou lista de adjacência com peso.

escolherGrafoDistancia ( ) - Apresenta opções de grafos a serem escolhidos para achar suas distancias e caminhos mínimos e passa para a função escolherParesTodosMST( ).

- **escolherParesTodosMST ( ) -** Apresenta a opção de buscar um grafo em algoritmos de distancia e caminho, chama as opções abaixo.
- **distanciaMSTBuscaEmLargura ( ) -** Realiza a distância e caminho usando busca em largura em pares de vértices de um grafo sem peso.
- **distanciaMSTDijkstra ( ) -** Realiza a distância e caminho usando o algoritmo de Dijkstra em pares de vértices de um grafo com peso.
- **distanciaTodosBuscaEmLargura ( ) -** Realiza a distância e caminho usando busca em largura em todos os vértices de um grafo sem peso.
- **distanciaTodosDijkstra ( ) -** Realiza a distância e caminho usando o algoritmo de Dijkstra em todos os vértices de um grafo com peso.

Resultados

Foram concluídas a parte 1 e as funções (1 e 2) da parte 2 do projeto, sendo devidas a entrega das duas últimas funções, os códigos e a biblioteca implementados foram coesos e é possível o uso para manipulação de grafos independentemente a falta das duas funções deixadas a fazer.