Projeto de Grafos
Relatório:

GABRIEL HENRIQUE CAMPOS SCALICI
9292970

O projeto realizado trata-se de um programa que utiliza a ideia de grafos para realizar o algortimo Floyd-Warshall para verificar a menor distância entre dois vértices, bem como o menor caminho para se percorrer no grafo de um vértice à outro.

TAD.

Primeiramente sobre o TAD GRAFO, foi feito uma lista de adjacência com os vetrices e com as arestas, onde cada nó aponta para o nó anterior e para o nó sucessor, em vista que foram realizadas muitas buscas para achar um determinado vértice e buscar determinada aresta, dessa forma a busca se torna mais simples.

Foram criadas três structs: Vértice, Aresta e Grafo para a realização desse projeto, além de todas as funções ditas obrigatórias no enunciado do mesmo.

Para cadastrar os componentes do grafo (vértices e arestas), são gerados automaticamente valores de identificação, começando em 1 e anotando o valor em um campo chamado “custo” dentro da struct.

Todas as operações de alteração do grafo foram feitas com base nas listas dos vértices e das arestas.

Os ponteiros para vértices (v1, v2) em cada aresta foram armazenados de forma que o v1 seja sempre menor que o v2.

MENU.

O menu foi feito na main de forma que realize a interação entre o usuário e a máquina.

Foi usado uma transformação simples, onde a operação digitada pelo usuário (duas letras), é transformada em um número inteiro, e tal número é usado no switch, esse por sua vez está dentro de um laço de repetição while que para somente quando for digitado “FM”, que é transformado para o número 11.

ALOCAÇÃO.

Todos os componentes que pertencem ao TAD ou ao algoritmo de busca pelo menor caminho, foram alocados de maneira dinâmica dentro do programa, para que possa funcionar com grandes valores de dados.

FLOYD-WARSHALL.

O FW foi implementado na main, pois não faz parte do TAD GRAFOS, ele apenas usa-o.

Para o funcionamento simples e legível foi divido em quatro funções, sendo elas a função que cria uma matriz de distância com base na quantidade n de vértices (N x N). Outra função que também utiliza o número de vértice para determinar o tamanho da matriz, para que possa ser anotado os vértices pais de cada outro vértice, dessa forma ficando fácil a verificação do menor caminho.

Um algoritmo principal que desenvolve a lógica, utilizando as duas matrizes anteriormente criadas e apenas a identificação de um vértice, dessa forma sem interferir no TAD. Atualizando a matriz de distância e a matriz de parentes.

Por último temos uma função recursiva que usa a matriz de parentes para imprimir para o usuário o menor caminho entre dois pontos.

A função de imprimir menor distância é apenas acessar a matriz, dessa forma é implementada na main.