SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE VIÇOSA · UFV
CAMPUS FLORESTAL

Trabalho prático 1 - Teoria e Modelo de Grafos

Leitura e Análise em Grafos

Angelo Cupertino Machado - 4695
Arthur Fernandes Bastos - 4679
Guilherme Augusto Schwann Wilke - 4685
Iury Martins Pereira - 4671

Florestal - MG
2022

Sumário

1. Introdução

O Trabalho apresentado tem como objetivo abordar todos os conhecimentos obtidos sobre grafos durante a primeira etapa do semestre através da implementação de uma biblioteca para manipulação e obtenção de informações a partir de grafos simples , não direcionados e ponderados.

Para o trabalho, foi designado o uso do site “PAAD - Grafos” para construir manualmente os grafos diferentes e gerar um arquivo .json, o qual será transformado em .txt e lido pela biblioteca criada pelos alunos. A biblioteca tem como funções:

- Retornar a ordem do grafo;
- Retornar o tamanho do grafo;
- Retornar os vizinhos de um vértice fornecido;
- Determinar o grau de um vértice fornecido;
- Retornar a sequência de graus do grafo;
- Determinar a excentricidade de um vértice;
- Determinar o raio do grafo;
- Determinar o diâmetro do grafo;
- Determinar o centro do grafo;
- Determinar a sequência de vértices visitados na busca em profundidade e informar a(s) aresta(s) que não faz(em) parte da árvore de busca em profundidade;
- Determinar distância e caminho mínimo;
- Determinar a centralidade de proximidade C de um vértice x, dada por:

$$C(x) = \frac{N-1}{\sum_y d(y,x)}$$

Onde N é o número de vértices do grafo e d(y,x) é a distância entre os vértices x e y. Esta medida é conhecida na área de redes sociais e refere-se a distância média de x a um vértice qualquer do grafo.

Por fim, a biblioteca deve possuir a opção de receber um arquivo em formato .txt de um grafo com mesma formatação das recebidas e transformá-lo em .json de modo a ser compatível com o lido pelo site.

2. Desenvolvimento

2.1 Leitura dos arquivos JSON e conversão para arquivo de texto

Para a leitura dos arquivos JSON nos quais se encontram os grafos foi implementada uma rotina em Python que converte-os em arquivos de texto utilizando a biblioteca json, assim como também pode converter de arquivo de texto para JSON. Em ambos os casos é preciso especificar qual arquivo deseja ler e qual arquivo deseja gerar dentro das funções.

2.2 Implementação dos grafos em C

Para representar e implementar os grafos em C utilizando o método de lista de adjacência foram implementadas 5 funções que, em geral, leem o arquivo de texto onde está localizada a representação do grafo, reproduzem a lista de adjacência de acordo com esse arquivo de texto e retornam a estrutura Grafo, a qual possui a quantidade de vértices existentes no grafo e uma lista de estruturas Vértice.

A estrutura Vértice possui o número identificador do vértice em questão e dois apontadores para o primeiro e último itens da lista encadeada que representa seus vizinhos. Todas as células da lista encadeada, chamadas de vertVizi, possuem o número identificador do vértice vizinho correspondente, o peso da aresta gerada entre o vértice representado pela estrutura Vértice e o vizinho correspondente na lista e um apontador para a próxima célula da lista encadeada.

2.3 Funções da biblioteca

2.3.1 ordemGrafo

A função ordemGrafo retorna a ordem do grafo que recebe como parâmetro utilizando a variável quantidadeDeVertices existente na estrutura Grafo.

2.3.2 tamanhoGrafo

A função tamanhoGrafo retorna o tamanho do grafo que recebe como parâmetro ao percorrer, para cada vértice do grafo, sua lista de vizinhos, contar quantos

vizinhos cada um deles tem e em seguida dividir esse número por 2, uma vez que uma aresta é representada 2 vezes na estrutura grafo, uma para cada vértice relacionado a essa aresta.

2.3.3 verticesVizinhos

Essa função retorna uma string que representa os vértices vizinhos de um dado vértice passado por parâmetro ao percorrer sua lista de adjacência e inserir os respectivos identificadores de cada célula na string.

2.3.4 verticeGrau

Essa função retorna o número de vértices vizinhos de um dado vértice passado por parâmetro ao percorrer sua lista de adjacência e contar quantas células ela possui.

2.3.5 sequenciaGrausGrafo

Essa função retorna uma string correspondente a sequência de graus do grafo ao coletar o grau de todos os seus vértices e inseri-los em uma lista, que será inversamente ordenada através do método QuickSort e transformada em string.

2.3.6 exentricidadeVertice

Essa função retorna a excentricidade de um vértice passado por parâmetro ao executar o algoritmo de Floyd Warshall para o grafo, analisar suas matrizes de caminhos e distâncias, caso o vértice tenha circuito negativo a função retorna um valor definido como -INFINITO, caso contrário, a função retorna o maior caminho entre o vértice em questão e qualquer outro vértice do grafo.

2.3.7 raioGrafo

Essa função retorna o raio do Grafo em questão ao recuperar a excentricidade de todos os vértices e retornar a menor excentricidade registrada.

2.3.8 diametroGrafo

Essa função retorna o raio do Grafo em questão ao recuperar a excentricidade de todos os vértices e retornar a maior excentricidade registrada.

2.3.9 centroGrafo

Essa função retorna uma string correspondente ao raio do Grafo em questão ao recuperar a excentricidade de todos os vértices e retornar todos os vértices com excentricidade igual ao raio do grafo.

2.3.10 buscaProfundidade

Essa função exibe na tela a sequência de vértices visitados dentro da busca em profundidade do grafo e informa a(s) aresta(s) que não faz(em) parte da árvore de busca em profundidade. Para tal, é recebido um vértice como parâmetro que é usado para a busca em profundidade, ao marcar todas as arestas possíveis e executar a busca, marcando quais arestas foram utilizadas para tal e quais não foram.

2.3.11 caminhoMinimoEntreVertice

Essa função retorna uma string correspondente ao caminho mínimo entre dois vértices passados por parâmetro ao recuperar a matriz caminho através do algoritmo de Floyd Marshall, que representa o caminho mínimo entre os vértices do grafo, e percorrer dentro da matriz, desde o vértice destino até o vértice inicial.

2.3.12 distanciaEntreVertice

Essa função retorna a distância entre dois vértices passados por parâmetro ao recuperar a matriz de distâncias dos vértices do grafo através do algoritmo de Floyd Marshall, e retornar seu valor correspondente.

2.3.13 centralidadeProximidade

Essa função retorna a centralidade de proximidade de um vértice passado por parâmetro ao dividir o número de vértices do grafo - 1 pelo somatório das distâncias do vértice em questão para todos os vértices do grafo utilizando o algoritmo de Floyd Warshall para recuperar suas distâncias.

3. Conclusão

Através desse trabalho, foi possível fixar conhecimentos adquiridos ao longo da primeira etapa da matéria de Teoria e Modelo de Grafos, assim como criar uma biblioteca funcional com o uso de diferentes métodos para manipular e recuperar informações de grafos simples.

Uma das decisões importantes para a execução do trabalho foi o uso do algoritmo de Floyd Warshall para recuperar o menor caminho entre vértices, uma vez que este também funciona com arestas de peso negativo, diferentemente dos outros algoritmos estudados.