

Resolução de Problemas com Grafos

PROJETO COLABORATIVO 1 (TDE3)

Prof. Vinícius M. A. Souza

Data da entrega: Verificar no AVA – **Atrasos não serão tolerados!**

O que é necessário para a entrega:

- Código-fonte em arquivos Python (extensão .py), **links não serão aceitos**;
- Não é preciso enviar o conjunto de dados.

Teste de autoria:

- Cada equipe deverá apresentar ao professor, em sala de aula e no dia da entrega, os resultados das funcionalidades implementadas, discutindo o código correspondente. A não realização da apresentação acarretará a atribuição da nota mínima (zero) para todos os integrantes da equipe.
- Cada membro da equipe deve apresentar ao menos uma funcionalidade diferente, mas todos devem demonstrar conhecimento sobre o código como um todo, evidenciando que o trabalho foi desenvolvido de forma colaborativa. A nota de cada funcionalidade será atribuída com base na apresentação realizada por um estudante da equipe.
- Caso um estudante não participe da apresentação ou demonstre desconhecimento de partes importantes do código, poderá receber a nota mínima (zero) individualmente, devido à falta de participação na implementação. Além disso, a funcionalidade que não for apresentada será considerada como não desenvolvida, o que poderá impactar a nota de toda a equipe.
- A nota de cada funcionalidade será atribuída com base na clareza, coerência e profundidade da explicação, na qualidade das respostas às perguntas do professor e na correção dos resultados apresentados.

Importante:

- Não serão considerados o uso de bibliotecas para grafos que já implementam as estruturas de dados ou funções solicitadas, assim como códigos e implementações de terceiros obtidos na Internet, incluindo ferramentas de Inteligência Artificial Generativa ou de outros colegas.

Equipes: até 5 estudantes, conforme os grupos indicados no Canvas.

Valor: 8 pontos (50% para o RA1 e 50% para o RA2)

Descrição do trabalho

O objetivo do trabalho consiste em desenvolver um analisador de contatos a partir da base de dados **Enron Email Dataset***. Nesse projeto procura-se explorar uma aplicação prática da teoria dos grafos que permite extrair informações úteis a partir da rede de contatos gerada com os e-mails presentes na base de dados.

* Endereço para o download da base de dados:

<https://drive.google.com/file/d/15vrDNLSYLvS4cvA0GILzeqx7SB4Mn6ud/view?usp=sharing>

Requisitos e Funcionalidades:

- 1) (1 ponto) A partir das mensagens de e-mail da base, construa um grafo direcionado considerando o remetente e o(s) destinatários de cada mensagem. O grafo deve ser ponderado, considerando a frequência com que um remetente envia uma mensagem para um destinatário. O grafo também deve ser rotulado, considerando como rótulo de cada vértice, o endereço de e-mail do usuário. Para demonstrar a criação do grafo, você deve salvar toda a lista de adjacências em um arquivo texto.
- 2) (1 ponto) Implemente métodos/funções para extrair as seguintes informações gerais do grafo construído:
 - a. O número de vértices do grafo (ordem);
 - b. O número de arestas do grafo (tamanho);
 - c. O número de vértices isolados;
 - d. Os 20 indivíduos que possuem maior grau de saída e os valores correspondentes (de maneira ordenada e decrescente de acordo com o grau);
 - e. Os 20 indivíduos que possuem maior grau de entrada e os valores correspondentes (de maneira ordenada e decrescente de acordo com o grau).
- 3) (2 pontos) Implemente uma função que verifica se o grafo é Euleriano (ou seja, que possui um ciclo Euleriano), retornando *true* ou *false*. Caso a resposta seja *false*, a sua função deve informar ao usuário todas as condições que não foram satisfeitas.
- 4) (2 pontos) Implemente um método que retorne uma lista com todos os vértices que estão localizados até uma distância D de um vértice N, em que D é a soma dos pesos ao longo do caminho mais curto entre dois vértices. A implementação deve ser eficiente o suficiente para lidar com grafos com milhares de vértices e arestas sem exceder limites razoáveis de tempo e memória.
- 5) (2 pontos) Implemente um método que calcule o diâmetro de um grafo, ou seja, o maior caminho mínimo entre qualquer par de vértices. O algoritmo deve retornar o valor desse maior caminho mínimo (diâmetro) e o caminho correspondente encontrado. Por simplicidade, desconsidere que o caminho mínimo entre dois vértices de componentes diferentes é infinito.