INSTITUTO FEDERAL
Sul de Minas Gerais
Campus Passos

TRABALHO PRÁTICO SOBRE GRAFOS

Curso:	Bacharelado em Ciência da Computação	**Ano:**	2020
Disciplina:	Teoria dos Grafos	**Período:**	5º
Professor:	Hugo Resende	**Data:**	04/03/2020
Trabalho individual ou em dupla			
OBSERVAÇÕES IMPORTANTES!		**Valor: 2,0 pontos**	

* Cópia de trabalhos = 0 (zero);
* Os trabalhos deverão ser enviados para *hugo.resende@ifsuldeminas.edu.br* até às 23h59 do dia 14/04/2020 com o assunto "[Teoria dos Grafos 2020] Trabalho Prático I";
* Mesmo que não consiga elaborar todo o trabalho, não deixe de enviar e apresentar o que desenvolveu;
* A apresentação dos trabalhos será no dia 15/04/2020;
* Trabalhos enviados após a data estipulada serão penalizados em 25% por dia de atraso.

MODELAGEM DE RELACIONAMENTOS ENTRE USUÁRIOS DE UMA REDE SOCIAL HIPOTÉTICA

A estrutura matemática combinatória grafos é conhecida na literatura computacional pelo seu poder de modelagem de diferentes tipos de cenários. Por exemplo, em uma rede social, os vértices de um dígrafo ponderado podem representar os usuários dessa rede, e os arcos (arestas direcionadas), relações de seguidores entre tais usuários (o peso de uma aresta, poderia ser um período de tempo associado entre dois usuários).

De modo a representar eficientemente as informações de um grafo, podem ser utilizadas várias estruturas computacionais. Dentre elas, destacam-se a matriz de adjacências (ou matriz de pesos, quando o grafo for ponderado), a lista de adjacências e a lista com adjacências em árvores AVL. Cada uma dessas estruturas possuem propriedades singulares, por exemplo, quando o grafo representado por tais são densos ou esparsos.

Com base nessas informações, o presente trabalho consiste no desenvolvimento de um programa em linguagens de programação C ou Java que modele um conjunto de relacionamentos de uma rede social hipotética, por meio das três estruturas representativas apresentadas no parágrafo anterior. Sumariamente, os vértices representam usuários que possuem *nome* e *idade*. A existência de um arco com peso *z* que parte de um vértice que representa um usuário *x* e incide em um outro vértice que representa um usuário *y* significa que *x* segue *y* e que *y* é seguido por *x* há um tempo *z* nessa rede social. Nesse sentido, o programa deverá fornecer recursos que realizem as ações das seguintes funções:

- *inicializarGrafos()*
  i. preenche a matriz de pesos com zeros, aloca uma posição de memória (posição *0*) para as listas de adjacência e adjacência em AVL e faz com o que seus conteúdos apontem para NULL;
  ii. não é necessário oferecer essa opção ao usuário;
  iii. poderão ser utilizados os índices da lista de *1* a *n*.

- *inserirUsuario()*
  i. libera, dinamicamente, uma linha e uma coluna da matriz de pesos para representar as relações do novo usuário, aloca uma posição de memória em cada uma das listas e realiza a inserção de tal usuário pelo nome (que será fornecido);
  ii. o programa deverá verificar se o usuário já está inserido e, caso positivo, retornar uma mensagem de erro.

- *inserirRelacao()*
  i. insere uma relação de "é seguidor de/seguido por" entre um par de usuários;
  ii. nessa operação de inserção, os usuários deverão estar previamente inseridos na rede;
  iii. deverão ser listados os nomes dos usuários da rede social para que o usuário do programa se guie na escolha;
  iv. a inserção da relação é concluída após a inclusão simultânea das informações nas três estruturas representativas;
  v. caso a relação já esteja inserida, deve-se oferecer a opção de atualizar a o período de tempo associado entre os dois usuários selecionados;

vi. no caso da representação na lista com adjacências em AVL, ao se inserir uma nova relação, é necessário verificar se as AVLs que representam os usuários foram desbalanceadas e, caso positivo, realizar as operações de rotacionamento de vértices (nós).
    - Revise os algoritmos de rotacionamento de árvores AVL e utilize os algoritmos já implementados em outras atividades.

- *listarSeguidores()*
  i. após o usuário do programa escolher um usuário cadastrado *x*, esta opção deverá listar todos os usuários os quais *x* segue e por quais usuários *x* é seguido, inclusive com os tempos relacionados à cada relação existente;
  ii. no caso da representação por lista com adjacências em AVL, fornecer ao usuário a opção de se utilizar algum dos algoritmos de percurso em árvores *in*-ordem, pré-ordem ou pós-ordem.

- *listarSeguidoresVelhos()*
  i. lista todos os usuários que são seguidos por usuários mais velhos, inclusive com os quantitativos associados a cada um deles.

- *atualizarRelacao()*
  i. ocorre similarmente à inserção de relações, porém a relação deverá estar previamente inserida.
  ii. caso a relação não esteja inserida deve-se oferecer essa opção ao usuário do programa.

- *removerUsuario()*
  i. remove um usuário previamente cadastrado na rede social, inclusive, com todas as suas relações;
  ii. caso o usuário não esteja cadastrado, exibir uma mensagem de erro.

- *removerRelacao()*
  i. remove uma relação previamente cadastrada na rede social;
  ii. caso algum elemento da relação a ser removida (vértice ou aresta) não esteja inserido, exibir uma mensagem de erro.

CRITÉRIOS DE AVALIAÇÃO E DETALHES ADICIONAIS

- Os pontos desta atividade serão distribuídos da seguinte maneira:
  - 1,6 pontos: implementação;
  - 0,4 pontos: apresentação do trabalho (avaliação individual).
- Apenas na função *listarSeguidores()* deverá ser oferecida uma opção para que o usuário do programa faça uma escolha, ao indicar qual estrutura representativa será utilizada para se exibir as informações. As demais funções deverão fazer alterações simultaneamente em todas as três estruturas;
- Para auxiliar no entendimento das informações elencadas neste trabalho e dos grafos, desenhe!
- Durante o desenvolvimento deste trabalho, como iremos lidar como muitas informações em diversos tipos de testes, evite o retrabalho! Elabore e utilize um arquivo com alguns usuários pré-cadastrados!
- Considere que não serão inseridos mais do que 50 (cinquenta) usuários, ou seja, serão necessárias estruturas com dimensões de, no máximo, esse valor de linhas e/ou colunas;
- Não haverá prorrogação do prazo de entrega do trabalho;
- Haverão aulas em laboratório de informática exclusivamente para o desenvolvimento deste trabalho.

Bom estudo!