



Campus Passos			
Curso:	Bacharelado em Ciência da Computação	Ano:	2022
Disciplina:	Teoria dos Grafos	Período:	4º
Professor:	Hugo Resende	Data:	26/08/2022
Trabalho individual ou em dupla			
OBSERVAÇÕES IMPORTANTES!		Valor: 2,0 pontos	
<ul style="list-style-type: none">* Cópia de trabalhos = 0 (zero);* Os trabalhos deverão ser enviados via Google Sala de Aula até às 23h59 do dia 27/09/2022;* Mesmo que não consiga elaborar todo o trabalho, não deixe de enviar e apresentar o que desenvolveu;* No caso do desenvolvimento do trabalho em dupla, os dois nomes deverão ser colocados no campo “Comentários”, no ato do envio;* As apresentações dos trabalhos ocorrerão nos dias 28/09/2022 e 30/09/2022;* Trabalhos enviados após a data estipulada serão penalizados em 25% por dia de atraso.			

UTILIZAÇÃO DE GRAFOS NA MODELAGEM DE UMA REDE SOCIAL

Diversos cenários reais são passíveis de modelagem por meio da estrutura matemática combinatorial grafos. Por exemplo, em uma malha rodoviária, os vértices de um grafo ponderado e não dirigido podem representar as cidades, e as arestas, possíveis rotas entre elas (com as suas respectivas distâncias). Na modelagem de aeroportos e voos, em um dígrafo ponderado, os vértices podem denotar a existência de um aeroporto, e as arestas, os custos associados de voo de um aeroporto a outro. Por fim, como um último exemplo, um dígrafo ponderado pode representar ainda as relações existentes entre seguidores de uma rede social, de modo que se existe uma aresta (i, j) com peso igual a 10, tal situação pode denotar que o usuário i segue o usuário j há 10 meses. Caso essa mesma aresta possuísse um sentido contrário, a situação seria referente ao usuário i ser seguido pelo usuário j há 10 meses.

Com base nessas informações, o presente trabalho consiste na implementação de um programa que possibilite um gerenciamento eficiente de usuários de uma rede social hipotética. De modo a modelar corretamente os dados do programa, deverá ser utilizado um grafo direcionado e ponderado, seguindo as diretrizes apontadas ao final do parágrafo anterior. O grafo utilizado como estrutura representativa de dados deverá ser representado computacionalmente por uma matriz de pesos.

RECURSOS DO PROGRAMA

O programa deverá fornecer recursos que realizem as ações das seguintes funções:

- *inicializaGrafo()*: atribui o valor 0 (zero) as posições da matriz de pesos;
 - Essa função deverá ser executada de forma automática, ou seja, não é necessário oferecer essa opção ao usuário;
 - Detalhe: os índices utilizados nas duas estruturas poderão ser $0 \dots n-1$ ou $1 \dots n$;
 - No dia da apresentação do trabalho o(s) integrante(s) deverá(ão) justificar a escolha.
 - De modo a parametrizar o programa, considere o número máximo de usuários em 100;
 - A estrutura inicial deverá ser estática, sendo liberadas posições de memória conforme demanda.
- *insereUsuario()*: reserva uma linha e uma coluna na matriz de pesos para representar uma eventual relação entre os usuários inseridos;
 - O usuário deverá ser inserido pelo nome;
 - O programa deverá verificar se o usuário já está inserido e, caso positivo, retornar uma mensagem de erro;
 - Deverá ser feito um controle de modo a manipular corretamente os nomes dos usuários com as suas respectivas linhas/colunas na matriz.

- *insereRelacaoSeguidor()*: insere uma relação entre um par de seguidores *i* e *j*, ou seja, *i* “segue” *j*, inclusive, com inserção de tempo (em meses);
 - Os usuários deverão estar previamente inseridos no grafo;
 - Caso contrário, retornar uma mensagem de erro.
 - Deverão ser listados os nomes dos usuários para que o usuário do programa se guie na escolha;
 - Caso a relação já esteja inserida, deve-se oferecer a opção de atualizar o tempo de relação entre os usuários selecionados.
- *segueQuem()*: após o usuário escolher um usuário, lista todos os usuários que são seguidos por ele, inclusive com os tempos associados;
- *quemMeSegue()*: após a escolha de um usuário, lista todos os usuários que o seguem, inclusive com os tempos associados;
- *atualizaRelacao()*: ocorre similarmente à inserção de relações, porém a relação a ser atualizada deverá estar previamente inserida;
 - Deverá ser fornecida uma opção ao usuário do programa, de modo que ele decida sobre qual tipo de relação atualizar (seguidor ou seguido por);
 - Caso a relação não esteja inserida, deve-se oferecer essa opção ao usuário;
 - Caso algum dos usuários, ou os dois, não esteja(m) inseridos, deverá ser oferecida essa opção ao usuário.
- *removeRelacao()*: com base na informação de um nome de usuário da rede social, realiza a deleção de uma relação;
 - Deverá ser fornecida uma opção ao usuário do programa, de modo que ele decida sobre qual tipo de relação atualizar (seguidor ou seguido por);
 - Caso o usuário não exista, emitir uma mensagem de erro.
- *calculaSeguidoresTotal()*: retorna o somatório de quantos seguidores cadastrados existem na rede social.

CRITÉRIOS DE AVALIAÇÃO E DETALHES ADICIONAIS

- Os pontos desta atividade serão, provavelmente, distribuídos da seguinte maneira:
 - 1,6 pontos: implementação;
 - 0,4 pontos: apresentação do trabalho (avaliação individual).
- Poderão ser utilizadas as linguagens de programação C ou Java para a implementação deste trabalho;
- O programa deverá possuir menu(s) interativo(s), de modo que o usuário possa se orientar da melhor maneira possível sobre as ações que irá realizar;
- Não haverá prorrogação do prazo de entrega do trabalho;
- Haverão várias aulas em laboratório de informática exclusivamente para o desenvolvimento deste trabalho;
- Outros detalhes e/ou observações poderão ser adicionados durante o desenvolvimento deste trabalho.