

Disciplina: **SCC0503 - Algoritmos e Estrutura de Dados II**
Professora: Elaine Parros Machado de Sousa
Estagiário PAE: Evandro Ortigossa

Trabalho 2 - Grafos

Instruções:

- **O trabalho deverá ser feito em duplas;**
- A entrega deve ser feita em **Atividade Trabalho Grafos** do Tidia até o dia **19/07/2020**, às 23:55h;
- Deve ser feito *upload* em um único arquivo compactado, contendo o **código fonte e um relatório pdf**. O nome do arquivo compactado deve ser formado pela sigla "Trab2_" concatenado a primeiro nome e último sobrenome do 1o aluno, conectado a primeiro nome e último sobrenome do 2o aluno. **Exemplo:** Trab2_ElaineSousa_EvandroOrgigossa.zip. **Somente 1** dos alunos do grupo deve fazer o *upload* da atividade!
- Faça um cabeçalho no início dos arquivos de código-fonte contendo as informações relevantes sobre seu trabalho, bem como nome completo e número USP dos integrantes do grupo, em comentário.
- Organize seu código utilizando comentários instrutivos.
- **Copiar o código em um PDF não é fazer um relatório, então não replique todo o código no relatório!** Em seu relatório, explique a lógica da sua solução, principalmente a estrutura lógica da implementação no grafo e as suas escolhas de implementação (algoritmos, por exemplo); justifique suas escolhas; explique o funcionamento da interface; e qualquer comentário que julgar necessário para entendimento da sua solução. Procure fazer um relatório de qualidade. Lembre-se que o relatório é o meio de comunicação entre você e quem irá avaliar seu trabalho;
- **LEIA este enunciado com atenção!** A falta de padrão será considerada.

Enunciado:

Sua tarefa é implementar um **TAD** (Tipo Abstrato de Dados) **Grafo** em C. Você deverá criar um grafo capaz de armazenar um conjunto de vértices interconectados por um conjunto de arestas, além de quaisquer outras informações necessárias. É sua tarefa modelar os atributos e funções/procedimentos do grafo de acordo com as especificações e requisitos do trabalho, fornecidos a seguir.

Aplique seu TAD Grafo na investigação do **mundo de Kevin Bacon** e para encontrar o **número de Kevin Bacon (KB)** de um/uma ator/atriz, determinado da seguinte maneira: um/uma ator/atriz que atuou com Kevin Bacon em um filme tem KB = 1; um/uma ator/atriz que nunca atuou com Kevin Bacon, mas atuou que alguém que já atuou com ele, tem KB = 2, e assim sucessivamente, sempre considerando o menor número possível de atores/atrizes na sequência. Se não for possível estabelecer nenhuma conexão entre o/a ator/atriz e Kevin Bacon, então KB = infinito. Por exemplo, Robert De Niro tem KB=1 pois atuou no filme "Sleepers" juntamente com Kevin Bacon. Elvis Presley tem KB=2 pois atuou com Edward Asner no filme "Change of Habit", o qual atuou no filme "JFK" com Kevin Bacon.

Modele o problema usando grafo: defina vértices e arestas, operações a serem

realizadas para resolver o problema, e características do grafo. Use o TAD para implementar a solução do problema proposto. Você está livre para escolher a representação por matriz de adjacências ou por listas de adjacências, desde que bem justificada para resolução do problema.

Especificações:

Você deve ler e processar um arquivo que contém as seguintes informações: uma lista de filmes com as respectivas listas de atores que trabalharam nestes filmes. O arquivo **input-top-grossing.txt** encontra-se no repositório do Tidia e foi construído a partir do Internet Movie Database (IMDb). Cada linha do arquivo é formada pelo nome de um filme, seguido pelo caracter "/" seguido pela lista de atores que atuaram neste filme (o nome de cada ator/atriz também está separado pelo caracter "/"). Por exemplo:

Picture Perfect (1997)/Aniston, Jennifer/Bacon, Kevin/Dukakis, Olympia Beach, The (2000)/DiCaprio, Leonardo/York, Daniel/Patarakijjanon, Patcharawan

Seu programa deve permitir que o usuário insira o nome de um ator e então exiba a sequência de filmes que leva este ator até Kevin Bacon. A saída do seu programa, para cada pesquisa inserida pelo usuário, deve ser formatada de acordo com o exemplo abaixo, que considerou como dado de entrada a atriz Dane, Cynthia:

"Dane, Cynthia" tem KB = 3
"Dane, Cynthia" atuou em "Solstice (1994)" com "Scott, Dennis"
"Scott, Dennis" atuou em "What About Bob? (1991)" com "Murray, Bill"
"Murray, Bill" atuou em "Wild Things (1998)" com "Bacon, Kevin"

Se o ator escolhido não existir, exiba uma mensagem informando tal fato.

Você também deve permitir que o usuário consulte a média de todos os números de Kevin Bacon do grafo (juntamente com o desvio-padrão). Seria o mundo de Kevin Bacon pequeno?

Requisitos:

Implemente uma Estrutura de Dados Dinâmica para grafos. Implemente um procedimento que leia e processe os dados a partir do arquivo fornecido e utilize as funções do seu TAD Grafo para construir e armazenar o grafo correspondente. Neste cenário, o presente trabalho deve apresentar o seguinte conjunto mínimo de operações:

- **endVertex(G, e)**: retorna referências para os dois vértices finais da aresta **e**;
- **opposite(G, v, e)**: retorna uma referência para o vértice oposto a **v** na aresta **e**;
- **areAdjacent(G, v, w)**: verdadeiro se os vértices **v** e **w** forem adjacentes, falso caso contrário;
- **insertVertex(G, o)**: insere um novo vértice isolado, armazenando nele o elemento **o**, e retorna uma referência para esse vértice;
- **insertEdge(G, v, w, o)**: insere uma aresta (**v, w**), armazenando nela o elemento **o**, e retorna uma referência para essa aresta;
- **replaceEdge(G, e, o)**: substitui o elemento da aresta **e** por **o**;
- **replaceVertex(G, v, o)**: substitui o elemento do vértice **v** por **o**;

- **edgeValue(G, e):** retorna o elemento armazenado na aresta **e**;
- **vertexValue(G, v):** retorna o elemento armazenado no vértice **v**.

Observações: (i) note que as operações descritas acima serão utilizadas como operações auxiliares para construção e acesso ao grafo para que o usuário possa explorar o mundo de Kevin Bacon (ou seja: use corretamente o conceito de TAD); (ii) não confunda o identificador de um vértice ou aresta com o conteúdo que pode ser armazenado naquele vértice ou aresta; (iii) você deve definir adequadamente o que será necessário armazenar em cada vértice e aresta para resolver o problema; (iv) as “referências” a vértices e arestas são ponteiros.

Interação com o usuário:

Seu programa deve permitir interação com o usuário a partir da entrada de dados padrão (teclado) pelo console (modo texto), e deve oferecer o seguinte conjunto mínimo de funcionalidades:

1. **Inicialização:** leitura e processamento dos dados para a montagem do grafo (inserção dos vértices e arestas). O grafo deve ser mantido em RAM;
2. **Número de Kevin Bacon (KB):** o usuário deve ser capaz de pesquisar o KB a partir da informação do nome de um ator/atriz;
3. **Mundo de Kevin Bacon:** o usuário deve ser capaz de consultar a média de todos os números de Kevin Bacon do grafo (juntamente com seu desvio-padrão);
4. **Finalizar a execução:** o usuário deve ser capaz de encerrar a execução do programa. Ao final da execução, o arquivo deve ser fechado e toda a memória alocada deve ser liberada pelo seu programa.

Observações importantes:

1. Implemente seu programa utilizando a linguagem **C padrão ANSI**. Caso estiver usando o compilador GCC, utilize a *tag* -ansi;
2. Utilize bibliotecas da linguagem **C ANSI**;
3. Você pode utilizar **seus** códigos desenvolvidos nas práticas de laboratório;
4. A clareza da sua implementação será avaliada;
5. Procure criar uma documentação útil e informativa (código e relatório), pois a qualidade de sua documentação será critério importante a ser considerado na correção do trabalho;
6. Não se esqueça de tratar warnings e as exceções que porventura possam ocorrer em seu programa. Vazamentos de memória, referência a valores de variáveis não inicializadas e outros problemas de programação serão levados em conta na avaliação;
7. Você tem tempo suficiente para desenvolver e testar todo o trabalho... **não deixe para a última hora**;
8. Variáveis globais não são recomendadas;
9. O **PAE (Evandro)** ficará online durante os horários de aula (email ou chat do hangouts - evortigosa@gmail.com) para apoio ao desenvolvimento e dúvidas. Note que o termo “apoio” não quer dizer que o PAE irá compilar seu código e encontrar os eventuais problemas. **IMPORTANTE:** o atendimento só será feito nesses horários - são quase 4 horas de atendimento por semana até a data de entrega. Lembrando:

3a - das 21h às 22:40h
6a - das 19h às 21:40h
De 30/06 até 17/07.

10. Lembrem-se da política sobre plágios que foi discutida na aula de apresentação da disciplina.

DICA para teste e debug: para testar se os procedimentos estão funcionando corretamente, faça uma primeira versão do programa com poucos elementos. Monte casos de testes que permitam testar todos os procedimentos e casos especiais. Isso facilitará o teste e debug.