Universidade Federal de Juiz de Fora
Departamento de Ciência da Computação
Disc. DCC059 – Teoria dos Grafos – Semestre 2022-1
Turma A
Profs. Stênio Sã

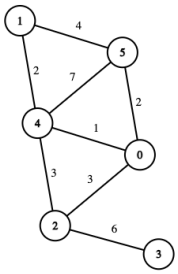
Especificação do Trabalho (Parte 1)

Este documento traz a especificação da **primeira parte** do trabalho prático da disciplina. Nesta primeira parte, não são necessários relatório e apresentação, apenas a implementação das funcionalidades definidas abaixo.

Objeto: desenvolver um Tipo Abstrato de Dados - TAD ou uma Classe que represente grafos simples, orientados e não orientados, ponderados e não ponderados (nos vértices e arestas) e implemente o conjunto de funcionalidades apresentadas a seguir, detalhadas em sala de aula.

Orientações:

- seu TAD ou Classe deve ser capaz de representar grafos utilizando **lista de adjacência**;
- o código deve ser desenvolvido em linguagem C ou C++ (padrão ANSI);
- Além do atendimento às funcionalidades, alguns dos elementos avaliados são a **clareza** e a **organização do código** (nomes de funções e variáveis, comentários que indiquem o propósito das principais funções e procedimentos, inclusive explicando o que são os parâmetros e o retorno, em caso de função);
- o programa principal que usará o TAD ou a Classe Grafo deve **ler os dados do grafo de entrada (direcionados ou não direcionados, ponderados ou não ponderados) a partir de arquivo texto**. O formato do arquivo dependerá da origem dos dados de entrada. Assim, cabe ao grupo ler o arquivo README que explica a semântica do arquivo de entrada ou, caso não haja este arquivo, o grupo deve ler o detalhamento do mesmo na fonte de dados, implementando conforme o caso. Entenda-se por formato do arquivo a estrutura em que os dados do grafo aparecem no texto do mesmo. Por exemplo, para algumas instâncias teste, o arquivo pode ser apresentado como segue no exemplo, onde se tem um grafo simples, não ponderado nos vértices e ponderado nas arestas, não direcionado. Neste exemplo, a primeira linha indica o número de vértices e as demais linhas indicam cada aresta e seu peso.




```
6
0 2 3
0 4 1
0 5 2
1 4 2
1 5 4
2 3 6
2 4 3
4 5 7
```

- a informação sobre o tipo de grafo, se direcionado ou não direcionado, deve ser passada ao programa por parâmetro via linha de comando (parâmetros da função *main*), sendo 0 (zero) para não direcionado e 1 (um) para grafos direcionados. Note que o TAD ou Classe deve prever a existência de duas formas de inclusão das adjacências, conforme cada caso.

- a informação sobre arestas ponderadas ou não deve ser passada ao programa por parâmetro via linha de comando, sendo 0 (zero) para não ponderado nas arestas e 1 (um) para grafos com peso nas arestas.

- a informação sobre a existência de pesos nos vértices deve ser passada ao programa por parâmetro via linha de comando, sendo 0 (zero) quando o grafo não é ponderado nos vértices e 1 (um) para grafos com pesos nos vértices.
- o nome do arquivo a ser lido deve ser informado ao programa via teclado para a função *main* (utilizar *int main* (*int argc, char ** argv*) para passar ao programa todas as informações necessárias ao seu funcionamento;
- cada grupo enviará um único e-mail contendo **APENAS** os arquivos fonte (extensão c, cc, cpp ou h) e os arquivos de entrada utilizados (quando o professor não os tiver encaminhado antes);
- o padrão para compilação a ser utilizado (ambiente Linux) será g++ *.c* - o execGrupoX. Onde "GrupoX" indica a qual grupo o trabalho se refere;
- o padrão para a execução a ser utilizado pelo professor será a linha abaixo, executada em ambiente Linux ou IOS:
- ./execGrupoX <arquivo_entrada> <arquivo_saida> <Opc_Direc> <Opc_Peso_Aresta> <Opc_Peso_Nos>, onde < arquivo_entrada> é o nome do arquivo que contém as informações do grafo, < arquivo_saida> é o arquivo onde será gravado o grafo armazenado na memória ao término do;
- o grupo deve enviar um **ÚNICO** arquivo compactado de nome Trabalho_GrupoX.**zip** para o e-mail indicado no plano de curso, cujo assunto da mensagem seja "Trabalho Grafos Grupo X – 2020-3", onde X indica o número do grupo.

Funcionalidades:

- O programa deve apresentar em **tela** a saída para as seguintes funcionalidades definidas nas saídas conforme os parâmetros:

a) **Grafo Interseção**: dados dois grafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2,E_2)$, o grafo interseção de G_1 com G_2 é dado por: $G_* = (V_1 \cap V_2, E_1 \cap E_2)$.
Saída: G_*. Note que a classe Grafo pode implementar este operador ou você pode ter este método definido como uma funcionalidade da interface com o usuário.

b) **Grafo União**: dados dois grafos $G_1 = (V_1, E_1)$ e $G_2 = (V_2,E_2)$, o grafo união de G_1 com G_2 é dado por: $G_+ = (V_1 \cup V_2, E_1 \cup E_2)$
Saída: G_+. Note que a classe Grafo pode implementar este operador ou você pode ter este método definido como uma funcionalidade da interface com o usuário.

c) **Grafo Diferença**: dados dois grafos $G_1 = (V, E_1)$ e $G_2 = (V,E_2)$, o grafo diferença de G_1 com G_2 é dado por: $G_- = (V, E_1 \setminus E_2)$. Note que operador exige que $E_1 \supseteq E_2$.
Saída: G_*. Note ainda que a classe Grafo pode implementar este operador ou você pode ter este método definido como uma funcionalidade da interface com o usuário.

d) **Rede Pert**: seja um grafo acíclico direcionado (GAD) representando o planejamento de execução de um projeto e onde cada nó representa etapa de um projeto e cada arco representa uma tarefa e indica a precedência das etapas conforme a cauda ou cabeça. Sabendo o tempo previsto para cada tarefa (arco), apresentar uma ordem de execução das tarefas que não implique no atraso para a conclusão do projeto,
Saída: a ordem das tarefas e o tempo total de execução do projeto;

- Todas as saídas das funcionalidades acima devem ser apresentadas na tela. Ao final da execução;
- Os itens a, b e c deverão ter saídas dadas na linguagem "dot" para geração de grafos para no software Graphviz (vide http://www.inf.ufes.br/~pdcosta/ensino/2018-2-estruturas-de-dados/material/Tutorial%20Graphviz.pdf).

Perguntas Frequentes

1. *Quantos membros um grupo pode ter?*

   O trabalho pode ter no **máximo 4 pessoas.** Mas, caso haja interesse de fazer o trabalho individualmente, você deve estar ciente de que, uma vez informado ao professor, não poderá integrar um outro grupo depois.

2. *O projeto poderá ter mais de um arquivo fonte (c, cc, cpp e h)?*

   Pode (e, para boa organização do código, deve). Como usual, a especificação do trabalho descreve somente a interface a ser implementada. A organização do projeto é livre.

3. *O que será levado em conta na correção?*

   Na correção serão levados em conta (entre outros) os seguintes elementos.

   1. **Interação com o professor;**
   2. **Conformidade com a especificação.**
   3. **Correção da implementação.**
   4. **Eficiência** da implementação.
   5. **Organização e clareza do código (nomes de funções e variáveis, comentários etc.).**

4. *Por que a especificação de como o programa será executado é importante?*

   Porque o trabalho entregue poderá ser pré-processado por um *script* que depende de a especificação de entrega ser corretamente observada.

5. *O que acontece se a especificação de execução do programa não for corretamente observada?*

   Seu trabalho só será corrigido quando houver tempo de fazer manualmente o pré-processamento. **Por isso, haverá desconto na sua nota, proporcional ao trabalho de pré-processamento que tenha que ser feito manualmente**.

6. *Meu trabalho tem um bug. O que vai acontecer com minha nota?*

   Haverá algum desconto, dependendo da gravidade do *bug*. Se o problema afetar alguma das funcionalidades requeridas, o desconto será proporcional ao que estiver faltando.

7. *Meu código não compila. Posso enviar assim mesmo?*

   Não serão avaliados trabalhos com erros de compilação. Por isso a importância de se usar apenas funções do padrão *Ansi*. Sugere-se que você compile e execute seu código no ambiente Linux conforme especificado neste documento.

8. *Tenho outra pergunta/dúvida a respeito do trabalho.*

   Procure os professores ou os tutores para tirar suas dúvidas no horário de atendimento ou durante as aulas.

Tenham todos um bom semestre e aproveitem a disciplina!

Profs. Stênio Sã