

Especificação de Trabalho

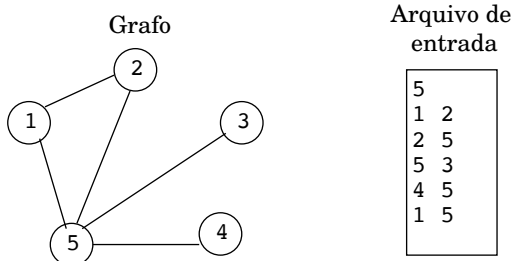
Este documento traz a especificação inicial das atividades avaliativas do tipo “Trabalho Prático” e servem de base para todas as atividades de implementação cobradas ao longo do curso da disciplina DCC059 - Teoria dos Grafos no semestre letivo 2020-3. Uma única data de entrega está estabelecida no Plano de Curso. Porém. Caso o grupo deseje, pode enviar antes do prazo.

Parte 1: Funcionalidades básicas em grafos simples

Objeto: desenvolver um Tipo Abstrato de Dados - TAD ou uma classe que represente grafos simples, não orientados e não ponderados e implemente o conjunto de funcionalidades apresentados a seguir, detalhadas em sala de aula. O desenvolvimento dos algoritmos aqui requeridos dá a base necessária para a resolução das questões das avaliações teóricas.

Orientações:

- seu TAD deve ser capaz de representar grafos utilizando **lista de adjacência**;
- o código deve ser desenvolvido em linguagem C ou C++ (padrão ANSI);
- 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 deve ler os dados de grafos simples (direcionados ou não direcionados) a partir de arquivo texto. O formato do arquivo dependerá da fonte dos dados de entrada. Assim, cabe ao grupo ler o arquivo README explicativo do formato do arquivo de entrada ou 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 nas arestas, e não direcionado. Neste exemplo, a primeira linha indica o número de vértices e as demais linhas indicam as arestas.



- a informação sobre o tipo de grafo, se direcionado ou não direcionado, deve ser passada ao programa por parâmetro via linha de comando, sendo 0 (zero) para não direcionado e 1 (um) para grafos direcionados. Note que o TAD 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 unico e-mail contendo **APENAS** os arquivos fonte (extensão c, cc, cpp e h) e os arquivos de entrada utilizados (quando o professor não os tiver encaminhado antes);
- O padrão para compilação a ser utilizado (ambiente Linux ou IOS) 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 em sala de aula, cujo assunto da mensagem seja "Trabalho Grafos Grupo X – 2020-3", onde X indica o número do grupo.

Nota: no arquivo zip deve ter incluso **APENAS** os arquivos com extensão c, cc, cpp ou h e os arquivos das instâncias usadas no experimento.

Funcionalidades da Parte 1:

- O programa deve apresentar em **tela** a saída para as seguintes funcionalidades:
 - a) Subgrafo induzido por um conjunto de vértices (parâmetro de entrada do método/função)
 - b) Caminho mínimo entre dois vértices (parâmetros) usando algoritmo de Dijkstra;
 - c) Caminho mínimo entre dois vértices (parâmetros) usando algoritmo de Floyd;
 - d) Árvore Geradora Mínima usando o algoritmo de Prim;
 - e) Árvore Geradora Mínima usando o algoritmo de Kruskal;
 - f) Busca em largura a partir de um dado nó (parâmetro);
 - g) Ordenação topológica de um DAG (Grafo Acíclico Direcionado);
- Todas as saídas das sete funcionalidades acima devem ser apresentadas na tela. Ao final da execução;
- Com exceção do item g, todas as saídas devem ser dadas no formato de grafos para o software Graphviz (vide <http://www.inf.ufes.br/~pdcosta/ensino/2018-2-estruturas-de-dados/material/Tutorial%20Graphviz.pdf>).

Parte 2: Problemas intratáveis em grafos

Objeto: estender seu Tipo Abstrato de Dados - TAD ou classe da parte 1 de forma a assegurar que represente grafos conforme que sejam instâncias do problema escolhido pelo grupo.

Para esta fase do trabalho, o TAD ou a Classe devem apresentar os seguintes algoritmos para o problema escolhido pelo grupo dentre os apresentados ao final deste documento:

- a) **Algoritmo guloso**
- b) **Algoritmo guloso randomizado;**

c) Algoritmo guloso randomizado;

Observações:

-
- a) Para esta fase do trabalho, o arquivo de entrada seguirá o formato definido pelo *benchmark* específico do problema escolhido;
 - b) Por se tratar de algoritmos estocásticos, convém a cada grupo procurar orientação do professor com antecedência para dirimir dúvidas quanto à análise adequada dos resultados.
 - c) O design de experimento a ser realizado nesta fase será detalhado em sala. Cabe a cada grupo avaliar o tempo que demanda o experimento de forma a assegurar-se de que o prazo para a entrega do relatório, do código e os arquivos de entrada e saída seja feita até a data estabelecida no Plano de Curso.
-

IMPORTANTE

- Ao final do trabalho (as duas partes), os seguintes materiais devem ser enviados ao professor para o e-mail stenio.soares@uff.br:
 - um relatório, conforme o modelo LaTeX (a ser apresentado no *Classroom* da disciplina) em mensagem cujo assunto deve ser “Relatório Grafos Grupo X – 2020-3”, onde X indica o número do grupo. O relatório deve informar as decisões de projeto e de implementação (fundamentadas nas dificuldades enfrentadas), além de exemplos das saídas do programa para as funcionalidades solicitadas. O relatório não deve ultrapassar 15 páginas, não deve ter listagem de código fonte e todo pseudocódigo inserido deve ter as linhas numeradas
 - Uma apresentação gravada em vídeo referente à parte 2 do trabalho, onde deve contar com a participação de **TODOS** os membros do grupo em que conste:
 - A descrição do problema;
 - Os algoritmos implementados para o mesmo;
 - Análise de resultados em comparação com algum trabalho da literatura correlata;
 - Conclusões
- Observação: o trabalho é em grupo e deve ser feito **sempre com a orientação do professor**. Os membros do grupo podem e devem trocar ideias entre si e levar ao professor as decisões de projeto para serem discutidas.

Temas para o trabalho da parte 2:

- 1 - Árvore Geradora com número mínimo de ramificações (árvore geradora de grau máximo mínimo) – para grafos não direcionados sem pesos nas arestas e nos vértices;
- 2 - Árvore Geradora Mínima Generalizada – para grafos não direcionados ponderados nas arestas;
- 3 - Árvore Geradora Mínima com Restrição de Grau – para grafos não direcionados ponderados nas arestas;
- 4 - Árvore Geradora Capacitada de Custo Mínimo – para grafos não direcionados ponderados nos vértices e nas arestas;
- 5 - Arborescência Mínima – para grafos direcionados ponderados nas arestas;
- 6 - Arborescência Capacitada de Custo Mínimo – para grafos direcionados ponderados nos vértices e nas arestas.

Perguntas Frequentes

1. *Quantos membros um grupo pode ter?*

O trabalho pode ter no máximo 5 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.

No caso de trabalhos individuais, o nome do arquivo a ser entregue deve ser fulano.zip, sendo que fulano deve ser substituído pelo nome do autor.

2. *Se eu fizer alguma parte do trabalho individualmente ou em grupo, terei que fazer o trabalho da fase correspondente à Unidade 6 da mesma forma (mesmo grupo)?*

Sim, pois a avaliação do trabalho considera a composição de todas as funcionalidades requeridas.

3. *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.

4. *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).
6. Competitividade dos resultados da parte 2 do trabalho;
7. Escrita do relatório.

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

Porque o trabalho entregue será pré-processado por um programa que depende de a especificação de entrega ser corretamente observada.

6. *O que acontece se a especificação de execução 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.**

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

Haverá algum desconto, dependendo da gravidade do *bug*. O desconto será menor se o *bug* for informado no relatório, indicando que você estava ciente do problema quando entregou o código.

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

Procure o professor para tirar suas dúvidas no horário de atendimento ou durante as aulas

Tenham todos um bom semestre e aproveitem a disciplina!

Prof. Stênio São Rosário Furtado Soares