



## Especificação de Trabalho Prático – Fase 1

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 2019-3.

**Objeto do trabalho:** desenvolver os Tipo Abstrato de Dados - TADs (ou classes) que implementem o conjunto de funcionalidades apresentados a seguir, detalhadas em sala de aula e todas aquelas que forem propostas ao longo do curso da disciplina.

### Orientações:

- seu TAD deve ser capaz de representar grafos direcionados e não direcionados utilizando lista de adjacência;
- o código deve ser desenvolvido em linguagem C ou C++;
- um dos elementos avaliados é clareza e 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 do grafo a partir de arquivo texto, conforme diretrizes a seguir:
  1. Para a leitura e escrita de grafos, vamos usar o seguinte formato:
  2. Uma string é uma sequência de até 1024 caracteres sem brancos, tabs ou newlines (isto é, sem "whitespace").
  3. Um grafo é dado por um arquivo onde cada linha tem zero, uma, duas ou três strings. Cada string representa um vértice (ou o peso de uma aresta entre dois vértices, caso haja três strings numa mesma linha).
  4. Pode-se ter arquivos onde ocorrem: uma única string em uma linha, que representa um vértice isolado; duas strings na mesma linha (separadas por "whitespace"), que representam dois vértices vizinhos entre si; três strings na mesma linha (separadas por "whitespace"), que representam dois vértices vizinhos e o peso da aresta (ou do arco) entre os mesmos.
  5. O arquivo [petersen.txt](#) tem um [grafo de Petersen](#) neste formato para servir como exemplo.
- 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;
- as informações referentes ao tipo de grafo ser direcionado ou não-direcionado e se é ponderado ou não ponderado nas arestas deve ser repassada ao programa via teclado para função main;
- para cada funcionalidade solicitada, seu programa deve apresentar uma opção em um menu apresentado no display, onde a opção 0 (zero) deve sair do menu e retornar ao prompt de comando.
- todo o código deve ser desenvolvido utilizando o padrão ANSI das linguagens C ou C++;
- o padrão de saída para cada funcionalidade será apresentado após a especificação de cada uma;
- cada grupo enviará ao e-mail informado 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>. Onde X é o número do grupo; < arquivo\_entrada> é o nome do arquivo que contém as informações do grafo; e < arquivo\_saida> é o arquivo onde será gravado o grafo armazenado na memória ao término do programa;
- o grupo deve enviar um único arquivo com extensão ZIP para o e-mail indicado em sala de aula, cujo assunto da mensagem seja “Trabalho Grafos Grupo X – 2019-3”, onde X indica o número do grupo.

**Nota: no arquivo zip deve ter incluso apenas os arquivos com extensão c, cc, cpp ou h.**

- ao final do trabalho (todas as fases), um relatório, conforme o modelo LaTeX (a ser apresentado enviado por e-mail) deve ser enviado ao professor para o e-mail indicado no item anterior em mensagem cujo assunto deve ser “Relatório Grafos Grupo X – 2019-3”, onde X indica o numero do grupo. O relatório deve informar as decisões de projeto e de implementação (fundamentadas nas dificuldades enfrentadas), além das saídas do programa para as principais 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.
- Observação: o trabalho é em grupo e deve ser feito **sempre sob a orientação do professor**, embora o grupo possa e deva procurar trocar ideias entre si e levar ao professor as decisões de projeto pra serem discutidas.
- **O TAD ou classe deve apresentar, para a primeira fase do trabalho (30% da nota = 9 pontos), as seguintes funcionalidades:**

#### Funcionalidades:

a - caminhamento em largura

a função deve receber como parâmetro o Id de um nó e imprimir o conjunto de arestas visitadas a partir do mesmo em um percurso em largura indicando, para cada uma, se trata-se ou não de uma aresta de retorno;

b - caminhamento em profundidade

a função deve receber como parâmetro o Id de um nó e imprimir o conjunto de arestas visitadas a partir do mesmo em um percurso em profundidade indicando, para cada uma, se trata-se ou não de uma aresta de retorno;

d – Componentes fortemente conexas (para grafos orientados)

para um dado grafo orientado G, a função não recebe nenhum parâmetro e deve imprimir o número de componentes fortemente conexas de G e, para cada componente, mostrar quais nós a compõem;

e - ordenação topológica (para grafo orientado)

para um dado grafo orientado acíclico, a função deve imprimir uma ordenação topológica do mesmo;

f – Dijkstra: caminho mínimo (para grafos orientados ou não orientados);

para um dado grafo (orientado ou não, ponderado ou não), a função deve receber o Id de dois vértices *u* e *v* e mostrar um caminho mínimo entre *u* e *v*, bem como o custo deste caminho calculado a partir do algoritmo de Dijkstra. O caminho a ser mostrado consiste na sequência de vértices entre *u* e *v* tal

que o somatório dos pesos das arestas entre vértices consecutivos na sequência seja mínima. Note que, para grafos não ponderados, um caminho mínimo entre dois vértices consiste numa sequência de vértices entre os mesmos com o menor número de arestas;

g - Floyd: caminho mínimo (para grafos orientados ou não orientados)

para um dado grafo (orientado ou não, ponderado ou não), a função deve receber o Id de dois vértices  $u$  e  $v$  e mostrar um caminho mínimo entre  $u$  e  $v$ , bem como o custo deste caminho calculado a partir do algoritmo de Floyd. O caminho a ser mostrado consiste na sequência de vértices entre  $u$  e  $v$  tal que o somatório dos pesos das arestas entre vértices consecutivos na sequência seja mínima. Note que, para grafos não ponderados, um caminho mínimo entre dois vértices consiste numa sequência de vértices entre os mesmos com o menor número de arestas;

h – Árvore Geradora Mínima (para grafos não orientados ponderados ou não) - Prim

para um dado grafo não orientado (ponderado ou não), utilizando o algoritmo de Prim, a função deve retornar um conjunto com  $n-1$  arestas que conecte todos os nós do grafo e cujo somatório dos pesos das arestas seja mínimo. No caso de grafos não ponderados, qualquer conjunto com  $n-1$  arestas que conecte o grafo é solução do problema;

i – Árvore Geradora Mínima (para grafos não orientados ponderados ou não) - Kruskal

para um dado grafo não orientado (ponderado ou não), utilizando o algoritmo de Kruskal, a função deve retornar um conjunto com  $n-1$  arestas que conecte todos os nós do grafo e cujo somatório dos pesos das arestas seja mínimo. No caso de grafos não ponderados, qualquer conjunto com  $n-1$  arestas que conecte o grafo é solução do problema;

j – Fecho triádico

Um fecho triádico ocorre em uma rede social se duas pessoas que tem um amigo em comum também são amigas entre si. O coeficiente de agrupamento (*clustering coefficient*) de uma rede é uma métrica que indica a frequência com este fenômeno ocorre na rede.

Dado um grafo  $G$ , chamamos de tríade um par de vértices  $\{u,v\}$  com um vizinho em comum  $w$ . Se os vértices  $u$  e  $v$  são vizinhos, dizemos que a tríade é *fechada* (este é um caso onde há um *fecho triádico*). Caso contrário, dizemos que a tríade é *aberta*. Note que para cada três vértices  $u$ ,  $v$  e  $w$  mutuamente adjacentes, existem três tríades fechadas. O *coeficiente de agrupamento* de um grafo é dado pelo número de tríades fechadas dividido pelo total de tríades (incluindo tríades abertas e fechadas). Convencionamos que o coeficiente de agrupamento de um grafo sem tríades é zero.

## Perguntas Frequentes

### 1. Quantos membros um grupo pode ter?

O trabalho pode ser feito em grupos de 4 ou 5 pessoas. Mas, caso haja interesse de fazer o trabalho individualmente, você deve estar ciente 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 este trabalho individualmente ou em grupo terei que fazer o trabalho da fase 2 da mesma forma (mesmo grupo)?

Sim.

### 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. Conformidade com a especificação.
2. Correção da implementação.
3. **Eficiência** da implementação.
4. Organização e clareza do código (nomes de funções e variáveis, comentários etc).

5. *Por que a especificação de entrega 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 entrega não for corretamente observada?*

Seu trabalho só será corrigido quando houver tempo de fazer manualmente o pré-processamento. **Por isso, haverá um severo 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.

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

Procure o professor para tirar suas dúvidas pessoalmente no horário de atendimento.