

Trabalho Prático (em grupo de 5 alunos)
Fase 1:

Objeto do trabalho: desenvolver um Tipo Abstrato de Dados – TAD ou uma classe que implemente o conjunto de funcionalidades apresentados a seguir e outras a serem requeridas ao longo do curso.

Orientações:

- seu TAD deve ser capaz de representar grafos (possivelmente digrafos, ponderados ou não) utilizando lista de adjacência;
- o código deve ser desenvolvido na linguagens C ou C++ e deve estar bem comentado (**código sem documentação será penalizado na nota**);
- o programa principal que usará o TAD deve ler os dados do grafo (ou digrafo) a partir de arquivo texto, conforme modelo indicado neste documento;
- para cada funcionalidade solicitada, seu programa deve apresentar uma opção em um menu apresentado no display;
- todo o código deve ser desenvolvido utilizando o padrão ANSI da linguagem C/C++;
- o padrão para a execução a ser utilizado pelo professor será a linha abaixo, executada em ambiente Linux ou IOS:

◦ ./grafosGrupoX <nome_arquivo_entrada>
<nome_arquivo_saida>

onde X é o número do grupo;
<nome_arquivo_entrada> é o nome do arquivo que contém as informações do grafo; e
<nome_arquivo_saida> é o arquivo onde

serão gravadas as saídas do programa para cada funcionalidade e o grafo armazenado na memória ao término do programa ou após o usuário escolher a opção de menu para salvar o grafo.

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 – 2018-2", onde X indica o número do grupo. Nota: no arquivo zip deve ter incluso apenas os arquivos com extensão c, cpp ou h.

- ao final do trabalho, um relatório, conforme o modelo LaTeX a ser apresentado em sala de aula, deve ser enviado ao professor para o e-mail indicado no item anterior em mensagem cujo assunto deve ser "Relatório Grafos Grupo X – 2017-3", 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 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**, sendo que os grupos podem (e devem) trocar ideias entre si e levar ao professor as decisões de projeto pra serem discutidas.

O TAD ou classe deve apresentar as seguintes funcionalidades:

a – leitura de arquivo (conforme entrada) e escrita em arquivo no mesmo modelo da entrada;

b – inclusão e exclusão de nó e de aresta;

c – retornar o grau de um dado nó informado pelo usuário (caso seja um dígrafo, informar grau de entrada e de saída);

d – verificar a k -regularidade do grafo (k informado pelo usuário);

e – informar a ordem do grafo;

f – mostrar a vizinhança aberta de um dado nó informado pelo usuário;

g – mostrar a vizinhança fechada de um dado nó informado pelo usuário;

h – verificar se o grafo é completo;

i – verificar se o grafo é bipartido;

j – apresentar a sequência de graus;

k – apresentar o fecho transitivo direto (para dígrafos)

l – apresentar o fecho transitivo indireto (para dígrafos)

m – apresentar uma ordenação topológica (para dígrafos)

n – apresentar uma árvore geradora mínima usando o algoritmo de Kruskal;

o – apresentar uma árvore geradora mínima usando o algoritmo de Prim;

p – apresentar o custo do caminho mínimo entre dois vértices informados pelo usuário usando o algoritmo de Dijkstra;

q – apresentar o custo do caminho mínimo entre dois vértices informados pelo usuário usando o algoritmo de Floyd;

3 – Formato do arquivo de entrada:

```
5
1 2 15
3 4 11
4 1 12
4 5 112
5 2 5
5 1 25
1 3 48
2 4 1
2 3 6
3 5 71
```

Neste exemplo, temos um grafo ponderado nas arestas, onde a primeira linha indica que o grafo tem 5 vértices. Cada uma das demais linhas traz os dois vértices que formam arestas, seguidos pelo peso da mesma.

Fase 2 (os mesmos grupos)

Na segunda fase do trabalho, cada grupo escolherá um problema da classe NP-Hard definido sobre a estrutura de um grafo e deverá implementar três abordagens heurísticas construtivas para o problema, conforme especificado nos itens abaixo:

- a) A1 : Algoritmo guloso;
- b) A2: Algoritmo guloso randomizado;
- c) A3 : Algoritmo guloso randomizado reativo.

Cada grupo selecionará um conjunto de 10 instâncias referenciadas na literatura e executará o seguinte experimento computacional:

- i) executar o algoritmo A1 uma vez para cada instância, anotando o valor da função objetivo do problema para a solução encontrada;
- ii) para cada instância, executar o algoritmo A2 20 (vinte) vezes para cada valor de alfa (0.1, 0.2 e 0.3), sendo que cada execução deve considerar o número máximo de iterações $\text{max_Itera} = 500$ (quinhentos), anotando o valor da função objetivo do problema para a solução encontrada. Assim, A2 terá, para cada instância, três versões de teste (A2 com alfa = 0.1, 0.2 e 0.3 e 1000 iterações), sendo que deverão ser feitas 20 (vinte) execuções por instância.
- iii) para o algoritmo A3, considerar o vetor de alfa = [0.05, 0.10, 0.15, ..., 0.50] e o número máximo de iterações max_Itera igual a 2000 (dois mil). Para cada instância A3 deverá ser executado 20 (vinte vezes), anotando o valor da função objetivo do problema para a solução encontrada.
- iv) escrever o relatório do trabalho em LaTeX, conforme modelo enviado, em que conste:

- 1 – Introdução;
- 2 – Algoritmos propostos;
- 3 – Experimentos computacionais;

Para tanto, deve-se comparar os resultados dos algoritmos propostos entre si e com o resultados obtidos pela abordagem da literatura.

- 4 – Conclusão

- 5 – Referência Bibliográfica

Qualquer dúvida, enviar e-mail ou procurar o professor da disciplina ou o tutor.

Juiz de Fora, 26 de julho de 2018.