



UNIVERSIDADE PRESBITERIANA MACKENZIE
Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos – Turma: 6G



Projeto da Disciplina
Aplicação para Resolução de Categorias de Problemas com Grafos
(Parte 1)

O objetivo deste projeto é desenvolver uma aplicação em grupo (máx. 3 alunos) que permita resolver uma categoria de problemas com grafos.

Para isso, considerar um problema do mundo real (do cotidiano ou do seu interesse) que contenha no mínimo (*) 50 vértices e no mínimo (**) 150 arestas que:

- tem elementos/objetos/pessoas/coisas e estabeleça uma relação entre eles;
- a seguir, propor um estudo de caso exemplo, contendo dados reais; e
- montar um grafo (direcionado ou não) que modela o problema fazendo uso da ferramenta Graph Online (<https://graphonline.ru/pt/>), tutorial em vídeo: <https://graphonline.ru/pt/help>).

* Uma quantidade de vértices menor do que essa somente será permitida se for validada com antecedência junto ao professor.

** Uma quantidade de arestas menor do que o estabelecido somente será permitida se for validada com antecedência junto ao professor.

Antes de começar a modelagem e implementação do problema selecionado, validar o Problema escolhido com o professor. Ressalto que, caso seja do interesse e um problema aceitável, este projeto poderá ser utilizado como base para continuidade no TCC (Trabalho de Conclusão de Curso) do grupo.

O problema selecionado deve conter objetos (elementos) e uma relação entre eles que permita a sua modelagem como um grafo, direcionado ou não direcionado, rotulado ou não. Os dados (vértices, arestas e rótulos (se necessário)) devem constar no arquivo “grafo.txt”, com formato similar ao descrito a seguir:

Tipo do Grafo

n

0 “Identificação do Rótulo 0” “peso do vértice 0”

1 “Identificação do Rótulo 1” “peso do vértice 1”

2 “Identificação do Rótulo 2” “peso do vértice 2”

...

m

Aresta_0 Peso_aresta_0

Aresta_1 Peso_aresta_1

Aresta_2 Peso_aresta_2

Aresta_3 Peso_aresta_3

Aresta_4 Peso_aresta_4

...

Onde:

- **Tipo do Grafo**, dependendo da modelagem proposta para o problema, tem o valor:
 - 0 – grafo não orientado sem peso;



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos – Turma: 6G

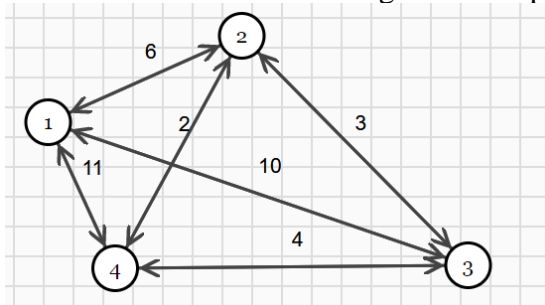


- 1 – grafo não orientado com peso no vértice;
 - 2 – grafo não orientado com peso na aresta;
 - 3 – grafo não orientado com peso nos vértices e arestas;
 - 4 – grafo orientado sem peso;
 - 5 – grafo orientado com peso no vértice;
 - 6 – grafo orientado com peso na aresta;
 - 7 – grafo orientado com peso nos vértices e arestas.
- n é o número de vértices e as “n” linhas seguintes representam o número vértice, apelido do vértice (se houver) e peso (caso haja);
 - m é o número de ligações (arestas) com os números correspondentes aos vértices que correspondem a cada aresta e o seu peso (se houver). Nas “m” linhas seguintes desse arquivo devem constar cada uma das arestas e o correspondente peso (caso haja).

Por exemplo: Considerando o grafo da Figura 1 (direcionado e com peso nas arestas), obtém-se o arquivo **grafo.txt** abaixo:

```
6
4
1
2
3
4
12
1 4 11
4 1 11
1 2 6
2 1 6
1 3 10
3 1 10
4 2 2
2 4 2
4 3 4
3 4 4
2 3 3
3 2 3
```

Figura 1. Grafo resultante da modelagem de um problema



Fonte: Autoria Própria.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos – Turma: 6G



Na programação do grafo, deve ser utilizado o código correspondente ao Grafo disponibilizado em aula, como matriz ou lista de adjacência, na linguagem C++ ou C ou Java ou Python e atualizá-lo com as mudanças necessárias para a implementação do seu projeto.

Na “Atividade Projeto 1” para o primeiro bimestre deve ser feito:

- 1) a definição do problema real selecionado pelo grupo, descrição textual (clara e com profundidade) desse problema (com imagem, se for pertinente), sua correspondente modelagem no *Graph online* e imagem do grafo relacionada;
- 2) a modelagem do problema no arquivo **grafo.txt**;
- 3) o desenvolvimento de uma aplicação contendo um menu de opções que permita
 - a) Ler dados do arquivo **grafo.txt**;
 - b) Gravar dados no arquivo **grafo.txt**;
 - c) Inserir vértice;
 - d) Inserir aresta;
 - e) Remove vértice;
 - f) Remove aresta;
 - g) Mostrar conteúdo do arquivo;
 - h) Mostrar grafo;
 - i) Apresentar a conexidade do grafo e o reduzido;
 - j) Encerrar a aplicação.

Onde:

- a) Deve ser lido o arquivo **grafo.txt** e montado o grafo correspondente como uma matriz ou lista de adjacências. Caso o grafo seja implementado como uma matriz de adjacência, sugere-se criar um vetor que armazena o rótulo do vértice e o seu apelido (localidade). A implementação do grafo como matriz ou lista de adjacências deve ser baseado NECESSARIAMENTE nas classes apresentadas em aula.
- b) Deve gravado o grafo da memória RAM para o arquivo **grafo.txt**, o formato do arquivo na gravação deve ser o mesmo da leitura.
- c) Permite inserir novos vértices juntamente com seus rótulos e peso (se for o caso).
- d) Permite inserir novas arestas com os respectivos pesos (se for o caso).
- e) Permite remover vértices juntamente com seu rótulo e peso (se for o caso). Cuidado: na remoção desse vértice pode ser necessário a remoção de arestas.
- f) Permite remover arestas com os respectivos pesos (se for o caso).
- g) Mostra o conteúdo atual do arquivo grafo.txt, ou seja, tipo do grafo, vértices e arestas em um formato visualmente compreensivo e atraente;
- h) Mostra o grafo como uma lista ou matriz de adjacência.
- i) Apresentar a conexidade do grafo: dependendo do tipo de grafo, mostrar se o grafo é conexo/desconexo – grafo não direcionado; e categoria C3, C2, C1 ou C0 – grafo direcionado. Além disso, se o grafo for direcionado, tendo por base o algoritmo FCONEX - que encontra as componentes fortemente conexas de um grafo direcionado, apresente o seu grafo reduzido.
- j) O programa deve ser encerrado.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos – Turma: 6G



Não se esquecer de colocar um título da aplicação (coerente com a categoria de problema que resolve) acima do menu de opções.

O grupo deve criar uma área no GitHub e publicar toda documentação produzida do projeto (incluindo o relatório); arquivo do grafo para o problema modelado; códigos fontes desenvolvidos; e link para o YouTube (que será gravado no 2º bimestre – continuação deste projeto, quando a aplicação for concluída).

Considerando a relevância dos Objetivos do Desenvolvimento Sustentável (ODS) para a humanidade, na tarefa de dissociar o crescimento econômico da pobreza, da desigualdade e das mudanças climáticas, destaque no relatório qual(is) objetivo(s) dos 17 do ODS o seu projeto contempla (pode ser um ou mais) e justifique cada um deles.

A saber, os 17 objetivos ODS são:



Para saber mais sobre os ODS, sugere-se consultar:

- Indicadores Brasileiros para os Objetivos de Desenvolvimento Sustentável: <https://odsbrasil.gov.br/>
- Guia Agenda 2030: www.guiaagenda2030.org/

Não se esquecer de colocar um título da aplicação (coerente com a categoria de problema que resolve) no seu menu de opções.

A título de informação, na “Atividade Projeto 2” para o segundo bimestre serão feitos eventuais inclusões de novas funcionalidades (opções do menu). Uma opção adicionada será a resolução do problema por uma das técnicas e algoritmos estudados.



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos – Turma: 6G



Observações:

1. O trabalho pode ser feito por grupos de até 3 pessoas.
 2. Um único aluno do grupo deverá publicar o trabalho no Ambiente Virtual.
 3. Deverá ser entregue um relatório com os resultados da **“Atividade Projeto 1”** deste projeto com base no Template disponibilizado contendo:
 - Dados dos integrantes do grupo (nome e RA).
 - **Título provisório da Aplicação (Essencial).**
 - a definição do problema real selecionado pelo grupo, descrição textual (clara e com profundidade) desse problema (com imagem, se for pertinente), sua correspondente modelagem no *Graph online* (ou outro software similar) e imagem do grafo relacionada.
 - Objetivos da ODS contemplados no seu projeto com detalhes e justificativa;
 - Printscreen de testes da execução das opções do menu. Ao menos 2 testes de cada opção.
 - O relatório deve conter ao seu final um Apêndice contendo o link do projeto no GitHub;
 - A área do GitHub deste bimestre, com acesso público, deve apresentar:
 - Relatório do Projeto;
 - Arquivo grafo.txt contendo o grafo modelado para o problema.
 - Os arquivos fontes, que devem contemplar:
 - um cabeçalho (comentário) com as identificações completas de todos os membros do grupo, síntese do conteúdo do arquivo e histórico das alterações realizadas, no formato: data da atualização, autor, breve descrição da atualização.
 - Documentação interna adequada e inclusão de comentários úteis e informativos.
 4. Junto ao relatório também devem ser entregues os códigos fontes e o arquivo txt. A entrega deve ser realizada na data limite 26 de setembro até as 23h59min.
 5. Deverá ser realizada uma apresentação do projeto no dia 30 de setembro, no horário da aula:
 - Na apresentação, o grupo deverá apresentar detalhes relacionados ao problema real selecionado, ao processo de construção do modelo, resultados obtidos e testes do menu de opções no tempo máximo de 5 (cinco) minutos.
 6. A apresentação em sala de aula compõe 20% da nota total do projeto.
- O projeto será avaliado de acordo com os seguintes critérios:
- Completude, clareza e ausência de erros de linguagem no relatório;
 - Funcionamento correto da Aplicação;



UNIVERSIDADE PRESBITERIANA MACKENZIE

Faculdade de Computação e Informática

Prof. Dr. Ivan Carlos Alcântara de Oliveira

Disciplina: Teoria dos Grafos – Turma: 6G



- O quão fiel é o programa quanto à descrição do enunciado;
- Endentação, comentários e legibilidade do código;
- Clareza na nomenclatura de variáveis e funções;
- Publicação do solicitado no GitHub;
- Apresentação realizada com clareza, conhecimento e cumprimento do tempo estabelecido.

Para auxiliar na documentação do código e entendimento do que é um programa com boa legibilidade siga as dicas apresentadas nas páginas abaixo:

- <http://www.ime.usp.br/~pf/algoritmos/aulas/layout.html>
- <http://www.ime.usp.br/~pf/algoritmos/aulas/docu.html>

Como este trabalho pode ser feito em **grupo**, evidentemente você pode “*discutir*” o problema dado com outros **grupos**, inclusive as “*dicas*” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento da sua aplicação.