

Avaliação Final

QXD0010 — Estrutura de Dados Avançada – Turma 01A

Prof. Atílio Gomes

27 de Outubro de 2020

Aluno: _____

Matrícula: _____

OBSERVAÇÃO: Logo abaixo, encontram-se três exercícios de programação, que você terá 24 horas para fazer.

Observação 1: Envie código compilável.

Observação 2: Provas copiadas receberão nota zero.

Prazo de Entrega: 28/10/2020 às 8h da manhã via Moodle, ou via email gomes.atilio@ufc.br (caso o Moodle esteja fora do ar).

-
1. Durante as aulas, aprendemos o que é a estrutura de dados **Grafos** e vimos que ela pode modelar situações reais. Vimos também que os grafos possuem duas representações principais no computador. Eles podem ser representados por **matriz de adjacência** ou por **listas de adjacências**.

Em aula, eu implementei em C++ a representação por matriz de adjacências usando o paradigma de orientação a objetos. Inicialmente, foi criada a classe abstrata **Graph** que possui as funções virtuais puras a serem obrigatoriamente implementadas pelas classes filhas (herdeiras) e também implementei a classe filha **Graphm**, herdeira de **Graph**. A classe filha **Graphm** é, na verdade, uma implementação que armazena o grafo como uma matriz de adjacência no computador. A implementação da classe que implementa o grafo por listas de adjacências ficou para casa.

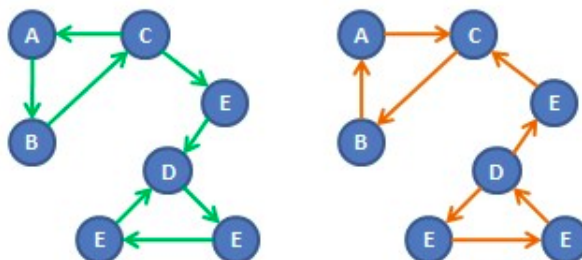
As classes **Graph** e **Graphm**, implementadas naquela aula, foram enviadas junto com esta prova.

A sua tarefa nesta questão é implementar a classe filha **Graphl**, que armazena o grafo como listas de adjacências. Juntamente com os demais arquivos foi enviado para vocês o arquivo **graphl.cpp** onde vocês devem implementar a classe **Graphl**.

Uma estrutura de dados só é utilizável (só presta) se ela funciona corretamente. Para saber se a estrutura de dados funciona corretamente não há outro jeito senão verificar que a implementação de cada função está correta. Uma forma empírica é testar as suas funções, ou seja, criar alguns testes e ver o que acontece e checar se o resultado obtido casa com o resultado esperado. Para isso, é preciso criar um programa cliente, ou seja, um arquivo **main.cpp** e criar um grafo e testar algumas de suas funcionalidades, de preferência exaustivamente. Para essa questão, crie um arquivo **main.cpp** e teste o seu grafo. Envie todos os arquivos mencionados acima como resposta desta questão.

Boa Prova!

-
2. (Exercício 22.1-3, do livro do Cormen, que está na bibliografia da disciplina)
- O **transposto** de um grafo direcionado $G = (V, E)$ é o grafo $G_T = (V, E_T)$, onde $E_T = \{(v, u) \in V \times V : (u, v) \in E\}$. Assim, G_T é G com todas as suas arestas invertidas. A figura abaixo mostra um grafo direcionado G à esquerda e o seu transposto G_T à direita.



Implemente algoritmos eficientes para calcular G_T a partir de G , para a representação por lista de adjacências e também para a representação por matriz de adjacências de G .

Dentro da classe `Graph` há o seguinte protótipo de função:

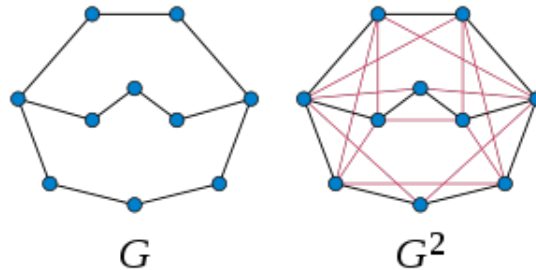
```
Graph* transposto();
```

Descomente essa linha quando você for fazer essa questão.

Dentro das classes `Graphm` e `Graphl` também há duas funções `transposto()`. O que esta questão pede é que você implemente essa função em cada uma dessas duas classes herdeiras. A função-membro `transposto()`, quando chamada por um objeto `g` do tipo `Graph`, devolve um ponteiro para um novo grafo que é o transposto do grafo `g` que invocou a função. Veja que, como estamos usando orientação a objetos, não é preciso passar o grafo `g` como parâmetro para a função. Desta forma, para criar o grafo transposto de `g`, é suficiente executar a seguinte linha no programa principal:

```
Graph *g;  
g = new Graphm(n);  
... (adiciona arestas no grafo g aqui)  
Graph* grafoTransposto;  
grafoTransposto = g.transposto();
```

-
3. (Exercício 22.1-5, do livro do Cormen, que está na bibliografia da disciplina)
O **quadrado** de um grafo direcionado $G = (V, E)$ é o grafo $G^2 = (V, E_2)$ em que $(u, v) \in E_2$ se e somente se G contiver um caminho que tenha no máximo duas arestas entre u e v . A figura abaixo mostra um grafo G e seu quadrado G^2 .



Implemente algoritmos eficientes para calcular G^2 a partir de G para uma representação por lista de adjacências e para uma representação por matriz de adjacências de G . Implemente os seus algoritmos como funções-membros das classes das questões anteriores, assim como foi feito para calcular o grafo transposto de G .