

CEFET-MG - Campus II
Departamento de Engenharia de Computação
Laboratório de Algoritmos e Estruturas de Dados II

Prática V

*Implementação em JAVA do TAD Matriz
de Adjacência e da Busca em Profundidade*

Alunos: Antônio Augusto Diniz Sousa - 201712040146
Professor orientador: Thiago de Souza Rodrigues

19 de Outubro
2018

CEFET-MG - Campus II
Departamento de Engenharia de Computação
Laboratório de Algoritmos e Estruturas de Dados II

Prática V

*Implementação em JAVA do TAD Matriz
de Adjacência e da Busca em Profundidade*

Relatório da prática V apresentado à Disciplina de
Laboratório de Algoritmos e Estruturas de Dados II
do Curso de Engenharia de Computação do Centro
Federal de Educação Tecnológica de Minas Gerais,
como requisito parcial para conclusão da disciplina.

Alunos: Antônio Augusto Diniz Sousa

Professor orientador: Thiago de Souza Rodrigues

19 de Outubro
2018

Sumário

1	Apresentação	1
2	Descrição de atividades	2
3	Resultados	2
4	Conclusão	4

1 Apresentação

A tarefa dessa prática foi testar o comportamento da TAD *Matriz de Adjacência* e do algoritmo de *Busca em Profundidade*, em dois grafos dados como exemplo, e, também, analisar se o mesmo possui ou não ciclos. Utilizou-se a implementação do Autor Ziviani, disponível nesse link ou acessado diretamente em <http://www2.dcc.ufmg.br/livros/algoritmos-java/implementacoes-07.php>.

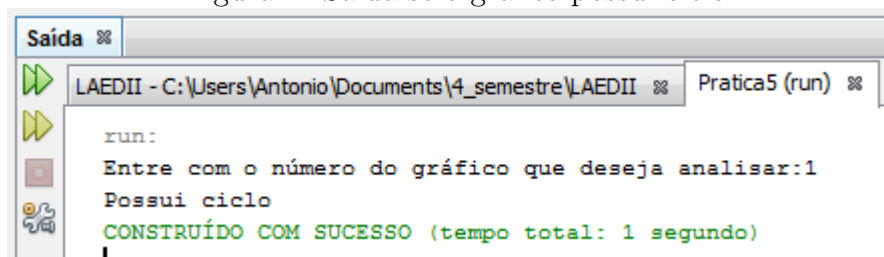
Boa parte do trabalho foi adiantado pela implementação citada anteriormente, porém foi necessário fazer uma pequena alteração na classe disponibilizada, para incluir a parte de análise de ciclo. Além disso, foi feito o código *main*, que possui a função principal de ler um arquivo que contém os dados do grafo a ser analisado, inserir em um objeto do tipo *Grafo* e rodar o método de análise.

Os resultados seguem adiante nesse relatório e os códigos fontes podem ser acessados clicando aqui ou acessando em <https://gitlab.com/antonioaads/LAEDII>.

2 Descrição de atividades

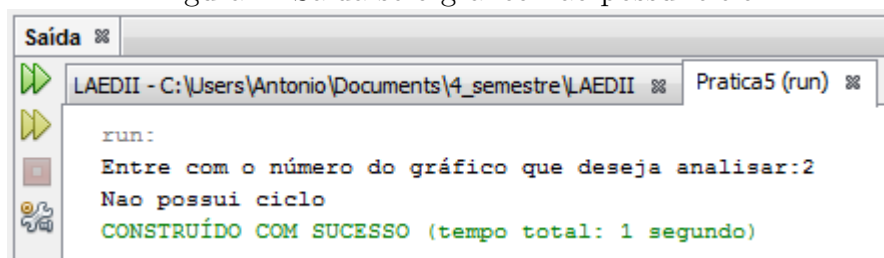
Após a execução do código, que não iremos tratar especificamente nesse relatório, pois o mesmo se encontra no repositório citado na introdução devidamente comentado e organizado, o programa retorna uma mensagem, informando se o grafo de entrada possui ou não ciclos, conforme Figura 1 e Figura 2.

Figura 1: Saída se o gráfico possui ciclo



```
run:
Entre com o número do gráfico que deseja analisar:1
Possui ciclo
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

Figura 2: Saída se o gráfico não possui ciclo



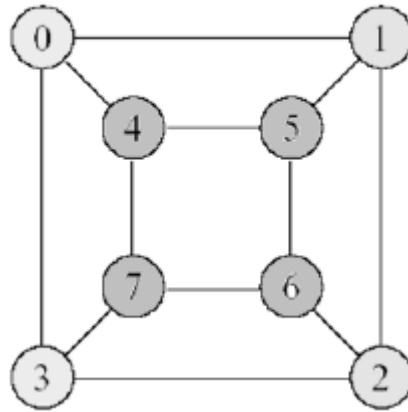
```
run:
Entre com o número do gráfico que deseja analisar:2
Nao possui ciclo
CONSTRUÍDO COM SUCESSO (tempo total: 1 segundo)
```

O início do programa, uma mensagem conforme essa é impressa: "Entre com o número do gráfico que deseja analisar:", na qual deverá informar o número do grafico que deseja analisar, que tem o nome conforme modelo: "grafon.txt", na qual **n**. representa qual grafo é analisado e será o valor que deverá digitar.

3 Resultados

O *print* da Figura 1 representa o resultado obtido ao rodar o grafo ilustrado pela Figura 3:

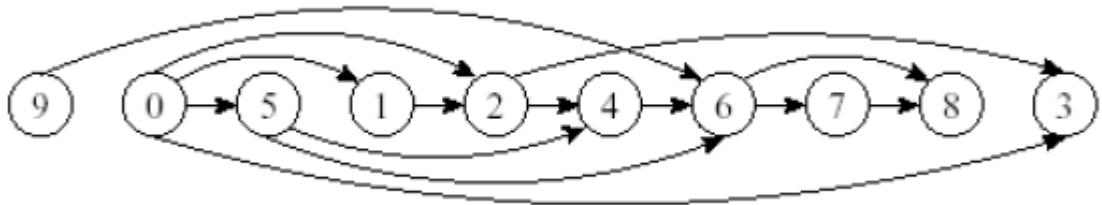
Figura 3: Grafo exemplo 1



Como pode ser observado facilmente, esse grafo possui diversos ciclos, justificando o resultado obtido.

O *print* da Figura 2 representa o resultado obtido ao rodar o grafo ilustrado pela Figura 4:

Figura 4: Grafo exemplo 2



Esse gráfico, diferente do exemplo 1, não possui ciclos, conforme resultado obtido.

4 Conclusão

A TAD Árvore de Adjacência é uma estrutura bem interessante e representa um grafo de maneira simples e de fácil acesso. Juntando essa estrutura com o algoritmo de busca em profundidade, temos um resultado muito interessante para diversos grafos, dentre os mais simples até os mais complexos, podendo tirar diversos dados, como o caso desse trabalho, a existência ou não de ciclos, ou até mesmo, analisar a existência de algum elemento, dentre outras aplicações.