

Felipe Augusto Ferreira de Castro **Matrícula:** 11711BCC033

Sarah Hanna VB Silva **Matrícula:** 11621BCC021

Renata Cristina Gomes da Silva **Matrícula:** 11721BCC012

Trabalho de Analise de Algoritmos

Universidade Federal de Uberlândia

2021

Sumário

Sumário	1
1	1
COLORAÇÃO DE GRAFOS	2
1.1	2
Pseudo-código	2
1.2	3
Estrutura de dados	3
1.3	3
Desenvolvimento do Trabalho	3
1.3.1	3
Grafos	3

1 Coloração de Grafos

1.1 Pseudo-código

O problema de coloração de grafos é um problema bastante discutido na literatura da área e possui vários algoritmos para solucioná-lo. Portanto, nesta seção apresentaremos o algoritmo usado em nosso trabalho para encontrar a disposição de cores e posteriormente discutiremos a eficiência do algoritmo. A seguir o algoritmo utilizado apresentado em pseudo-código.

Algorithm 1: Coloração de Grafos

Input: grafo

Output: Lista com as cores de cada vertice

declare uma variavel flag;

faça flag = verdadeiro;

atribua a cor 0 para todos os vertices do grafo;

primeiro vertice do grafo;

while *não atribuido uma cor a todos os vertices* **do**

while *para todos os vertices adjacentes* **do**

 olhe a cor do vertice adjacente;

if *possui a mesma cor que o vertice adjacente* **then**

 faça flag = falso;

 pare o laço;

else

end

end

if *flag = verdadeiro* **then**

 | proximo vertice;

else

 faça flag = verdadeiro;

 some 1 a cor deste vertice;

end

end

Explicando de maneira mais informal o algoritmo se resume em alguns passos:

- Para cada vértice v do grafo G vamos olhar as cores de seus vertices adjacentes;
- vamos avançando da lista de cores ate encontrar uma cor a qual não foi atribuida a nenhum vértice adjacente a v ;

- atribuímos a cor encontrada ao vértice v ;

1.2 Estrutura de dados

Visto o algoritmo para coloração apresentado na seção anterior foi decidido estruturar o grafo de maneira a facilitar encontrar os vértices adjacentes de cada vértice v do grafo G . Desta forma, cada vértice é uma estrutura que possui duas informações:

- valor da cor atribuída ao vértice;
- uma lista de identificadores dos vértices adjacentes;

A identificação do vértice adjacente vd é feita com um indexador da posição de vd na lista de vértices do grafo G .

A Estrutura do grafo é constituída de duas informações:

- Lista de vértices presentes no grafo(o indexador do vértice nessa lista é o identificador do vértice);
- quantidade de arestas presentes no grafo;

Utilizando desta estrutura encontrar os vértices adjacentes de um vértice v se resume a apenas percorrer uma lista, excluía a necessidade de verificar se o vértice vd é adjacente a v .

1.3 Desenvolvimento do Trabalho

O programa foi desenvolvido na linguagem Python devido a facilidade de encontrar ferramentas prontas para manipular estruturas de dados e o conhecimento prévio que os autores deste trabalho tinham sobre a linguagem. Além disso, o ambiente de desenvolvimento usado foi o Visual Studio Code.

1.3.1 Grafos

Quanto aos grafos, foram utilizados 5 grafos, no formato DIMACS, os quais serão apresentados a seguir.

1. Grafo 1

vertices: 10 , arestas: 15, a 0 1,
a 0 2,
a 0 3,
a 1 4,
a 1 8,

a 2 6,
a 2 7,
a 3 5,
a 3 9,
a 4 5,
a 4 7,
a 5 6,
a 6 8,
a 7 9,
a 8 9;

Fonte: O enunciado deste trabalho

2. Grafo 2

vertices: 9, arestas: 14,

a 0 3,
a 0 1,
a 1 2,
a 1 3,
a 2 4,
a 3 4,
a 3 6,
a 3 7,
a 4 5,
a 4 7,
a 4 8,
a 5 8,
a 6 7,
a 7 8;

Fonte: <https://coloringbee.blogspot.com/2018/09/coloring-graph-example.html>