

Relatório gerado a partir da análise do algoritmo de **Coloração de grafos**

Nome: Fabricio Monteiro de Aquino 11221BCC026

Nome: John Vitor da Silva Cunha 11821BCC005

Nome: Alexandre Albernaz Leite Lemos 12011BCC043

Nome: Laura Lorena Melo Amaral 11911BCC046

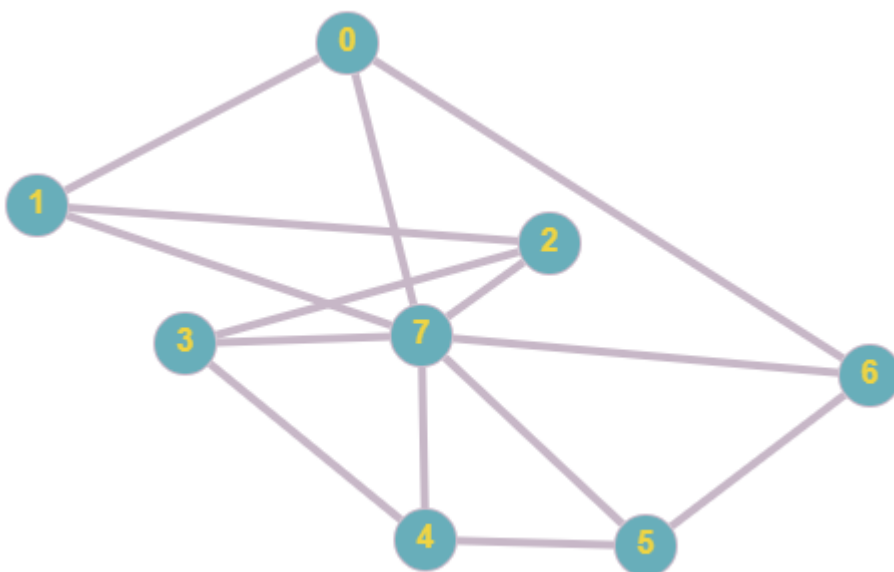
1) Linguagem utilizada:

_____No desenvolvimento do algoritmo foi utilizada a linguagem de programação JAVA. A linguagem foi escolhida devido a maior familiaridade que o grupo possui com ela, pois além de sua vasta utilização no mercado é bastante utilizada no meio acadêmico. Além disso ela é utilizada nas disciplinas de programação orientada a objetos 1 e 2 o que gera uma maior facilidade ao utiliza-la. Outro fator foi sua extensa e conhecida biblioteca visto que utilizamos funções contidas na biblioteca na geração das listas encadeadas para o grafo.

2) Grafos gerados, Matriz de adjacência, dimensão de cada grafo e saída do algoritmo para cada grafo

Foram gerados os seguintes grafos para este experimento:

Grafo 1:



Matriz de adjacência do grafo 1:

```
mat1= 0 1 0 0 0 0 1 1
      1 0 1 0 0 0 0 1
      0 1 0 1 0 0 0 1
      0 0 1 0 1 0 0 1
      0 0 0 1 0 1 0 1
      0 0 0 0 1 0 1 1
      1 0 0 0 0 1 0 1
      1 1 1 1 1 1 1 0
```

A dimensão de um grafo é o seu número de arestas. Assim a dimensão do grafo 1 é: 14.

Saida do algoritmo para o grafo 1:

Resultado para o grafo 1:

Vértice 0 ==> Cor: 0

Vértice 1 ==> Cor: 1

Vértice 2 ==> Cor: 0

Vértice 3 ==> Cor: 1

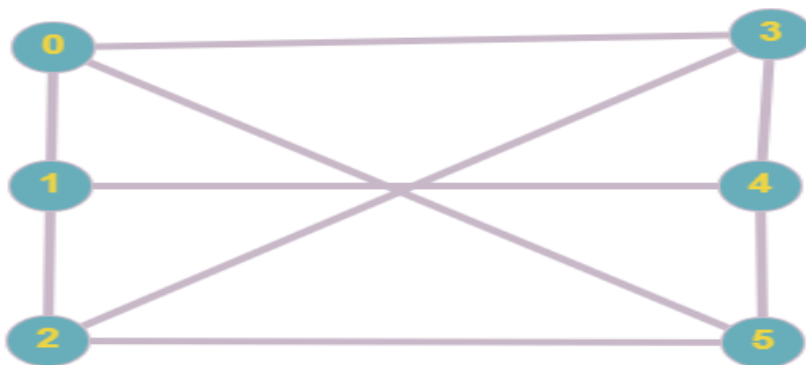
Vértice 4 ==> Cor: 0

Vértice 5 ==> Cor: 1

Vértice 6 ==> Cor: 2

Vértice 7 ==> Cor: 3

Grafo 2:



Matriz de adjacência do grafo 2:

```
mat2 = 0 1 0 1 0 1
        1 0 1 0 1 0
        0 1 0 1 0 1
        1 0 1 0 1 0
        0 1 0 1 0 1
        1 0 1 0 1 0
```

A dimensão do grafo 2 é: 9

Saida do algoritmo para o grafo 2:

Resultado para o grafo 2:

Vértice 0 ==> Cor: 0

Vértice 1 ==> Cor: 1

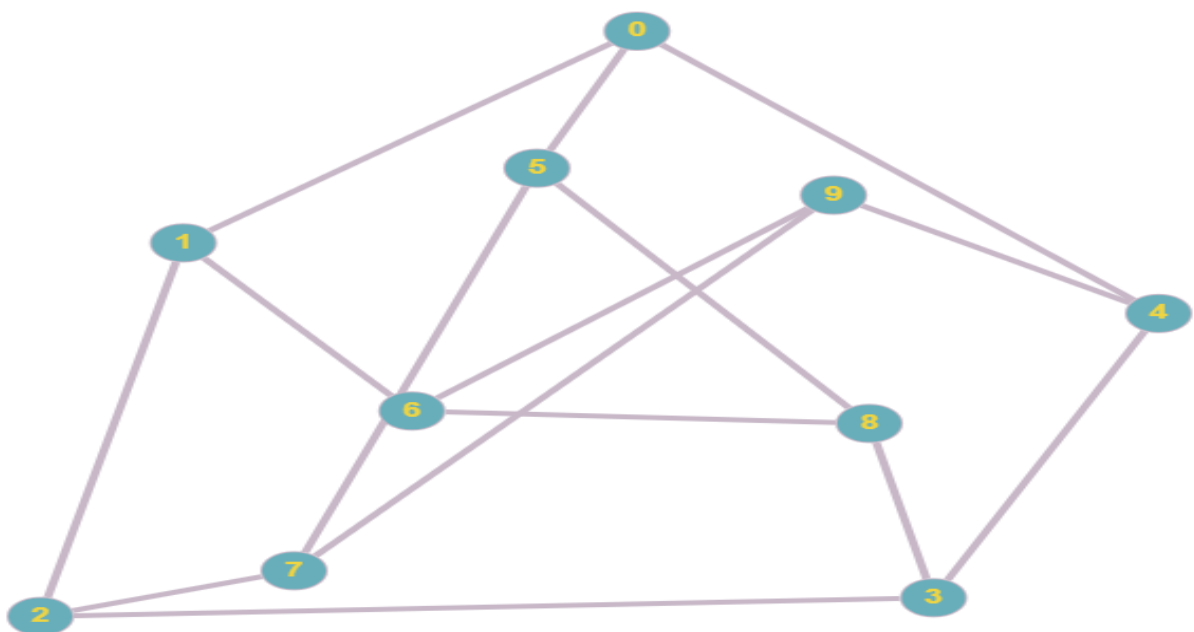
Vértice 2 ==> Cor: 0

Vértice 3 ==> Cor: 1

Vértice 4 ==> Cor: 0

Vértice 5 ==> Cor: 1

Grafo 3:



Matriz de adjacência do grafo 3:

```
      0 1 0 0 1 1 0 0 0 0
      1 0 1 0 0 0 1 0 0 0
      0 1 0 1 0 0 0 1 0 0
      0 0 1 0 1 0 0 0 1 0
mat3 = 1 0 0 1 0 0 0 0 0 1
      1 0 0 0 0 0 0 1 1 0
      0 1 0 0 0 0 0 0 1 1
      0 0 1 0 0 1 0 0 0 1
      0 0 0 1 0 1 1 0 0 0
      0 0 0 0 1 0 1 1 0 0
```

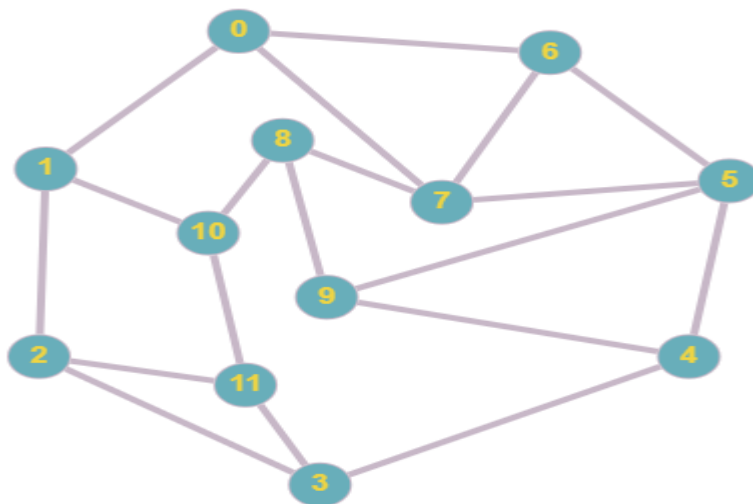
A dimensão do grafo 3 é: 16.

Saida do algoritmo para o grafo 3:

Resultado para o grafo 3:

```
Vértice 0 ==> Cor: 0
Vértice 1 ==> Cor: 1
Vértice 2 ==> Cor: 0
Vértice 3 ==> Cor: 1
Vértice 4 ==> Cor: 2
Vértice 5 ==> Cor: 1
Vértice 6 ==> Cor: 0
Vértice 7 ==> Cor: 2
Vértice 8 ==> Cor: 2
Vértice 9 ==> Cor: 1
```

Grafo 4:



Matriz de adjacência do grafo 4:

```

      0 1 0 0 0 0 1 1 0 0 0 0
      1 0 1 0 0 0 0 0 0 0 1 0
      0 1 0 1 0 0 0 0 0 0 0 1
      0 0 1 0 1 0 0 0 0 0 0 1
      0 0 0 1 0 1 0 0 0 1 0 0
mat4 = 0 0 0 0 1 0 1 0 0 1 0 0
      1 0 0 0 0 1 0 1 0 0 0 0
      1 0 0 0 0 0 1 0 1 0 0 0
      0 0 0 0 0 0 0 1 0 1 1 0
      0 0 0 0 1 1 0 0 1 0 0 0
      0 1 0 0 0 0 0 0 1 0 0 1
      0 0 1 1 0 0 0 0 0 0 1 0

```

A dimensão do grafo 4 é: 19.

Saida do algoritmo para o grafo 4:

Resultado para o grafo 4:

Vértice 0 ==> Cor: 0

Vértice 1 ==> Cor: 1

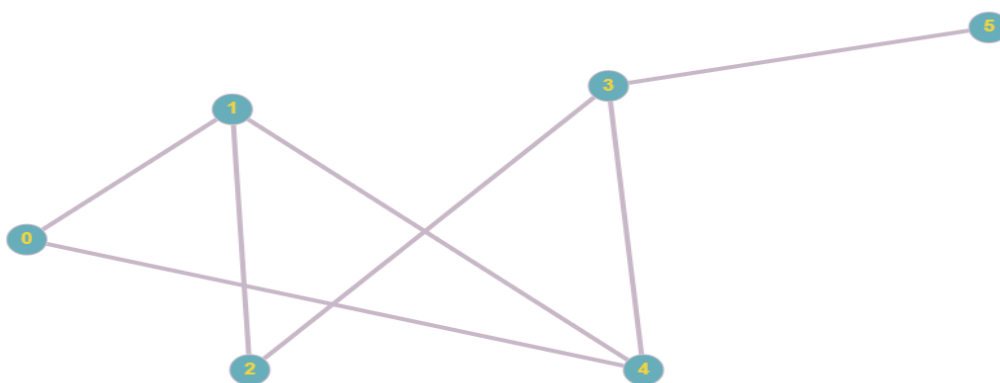
Vértice 2 ==> Cor: 0

Vértice 3 ==> Cor: 1

Vértice 4 ==> Cor: 0

Vértice 5 ==> Cor: 1
Vértice 6 ==> Cor: 2
Vértice 7 ==> Cor: 1
Vértice 8 ==> Cor: 0
Vértice 9 ==> Cor: 2
Vértice 10 ==> Cor: 2
Vértice 11 ==> Cor: 3

Grafo 5:



Matriz de adjacência do grafo 5:

```
mat5 = 0 1 0 0 1 0
        1 0 1 0 1 0
        0 1 0 1 0 0
        0 0 1 0 1 1
        1 1 0 1 0 0
        0 0 0 1 0 0
```

A dimensão do grafo 5 é: 7.

Saida do algoritmo para o grafo 5:

Resultado para o grafo 5:

Vértice 0 ==> Cor: 0
Vértice 1 ==> Cor: 1
Vértice 2 ==> Cor: 0

Vértice 3 ==> Cor: 1

Vértice 4 ==> Cor: 2

Vértice 5 ==> Cor: 0

3)Estratégia de programação utilizada:

No desenvolvimento deste projeto o algoritmo utilizado para gerar a coloração utiliza a estratégia gulosa. Pois o algoritmo pega sempre a primeira cor disponível sem analisar um histórico ou reconsiderar a escolha.

4)Tabela com o desempenho do algoritmo aplicado a cada grafo em cada execução

| Execução | Grafo1 | Grafo2 | Grafo3 | Grafo4 | Grafo5 |
|----------|------------|------------|------------|------------|------------|
| 1 | 0.007411 | 0.002586 | 0.004531 | 0.004248 | 0.00571701 |
| 2 | 0.007251 | 0.002538 | 0.010004 | 0.006618 | 0.003771 |
| 3 | 0.008616 | 0.005724 | 0.009887 | 0.00593499 | 0.00791499 |
| 4 | 0.00773101 | 0.00686401 | 0.00704399 | 0.01002099 | 0.006096 |
| 5 | 0.00764501 | 0.00265499 | 0.00608801 | 0.007448 | 0.00799899 |
| 6 | 0.008362 | 0.002886 | 0.005157 | 0.011531 | 0.004182 |
| 7 | 0.008333 | 0.003001 | 0.00679 | 0.007243 | 0.00713599 |
| 8 | 0.008374 | 0.038282 | 0.01495901 | 0.01952499 | 0.00685199 |
| 9 | 0.007675 | 0.00356599 | 0.006962 | 0.008213 | 0.014869 |
| 10 | 0.013789 | 0.00384401 | 0.00591499 | 0.006135 | 0.007391 |
| 11 | 0.00813 | 0.003863 | 0.01080399 | 0.006137 | 0.00700999 |
| 12 | 0.006033 | 0.002524 | 0.004635 | 0.00440801 | 0.003532 |
| 13 | 0.008619 | 0.00417101 | 0.005486 | 0.00713601 | 0.004284 |

| | | | | | |
|----|------------|------------|------------|------------|------------|
| 14 | 0.00828299 | 0.003481 | 0.005916 | 0.00735001 | 0.00735001 |
| 15 | 0.008736 | 0.005551 | 0.00534701 | 0.00940399 | 0.004246 |
| 16 | 0.010318 | 0.002456 | 0.005144 | 0.00715099 | 0.004327 |
| 17 | 0.00813799 | 0.00254399 | 0.00453399 | 0.00738301 | 0.00417701 |
| 18 | 0.007576 | 0.00235499 | 0.004789 | 0.006588 | 0.00423799 |
| 19 | 0.008364 | 0.002754 | 0.004456 | 0.007685 | 0.004096 |

| | | | | | |
|--------------------|-----------------|----------------------|----------------------|----------------------|----------------------|
| 20 | 0.00801799 | 0.00351199 | 0.005174 | 0.006518 | 0.00334899 |
| 21 | 0.00864499 | 0.002376 | 0.00448801 | 0.00682701 | 0.004074 |
| 22 | 0.01111301 | 0.00349401 | 0.00517299 | 0.007532 | 0.006618 |
| 23 | 0.00752299 | 0.004878 | 0.006272 | 0.009582 | 0.006415 |
| 24 | 0.008891 | 0.00532199 | 0.00687799 | 0.007863 | 0.00637101 |
| 25 | 0.011448 | 0.00608899 | 0.007713 | 0.00836701 | 0.007871 |
| 26 | 0.008531 | 0.00259599 | 0.005342 | 0.007126 | 0.004333 |
| 27 | 0.008187 | 0.00342501 | 0.00631501 | 0.00919 | 0.004187 |
| 28 | 0.009381 | 0.003952 | 0.006099 | 0.007103 | 0.004319 |
| 29 | 0.007667 | 0.004157 | 0.005496 | 0.006066 | 0.005932 |
| 30 | 0.008086 | 0.002649 | 0.004958 | 0.012508 | 0.006948 |
| | | | | | |
| TOTAL | 0.25687398 | 0.14409497 | 0.19235598 | 0.23884101 | 0.1756097 |
| MÉDIA =total/30 | 0.008562466 | 0.004803165 | 0.006411866 | 0.007961367 | 0.585365667 |
| Desvio Padrão | 0.0014370073023 | 0.006333265 61772 | 0.0022621786 8941 | 0.002770007 51167 | 0.0022399715 2535 |

Para o cálculo do tempo em cada execução do algoritmo foi utilizado o método estático `nanoTime()` da classe `System` o método foi aplicado apenas na parte do algoritmo que faz a coloração pois é ele o responsável pela estratégia de colorir o grafo, a função devolve os tempos em nanossegundos. Para este experimento os tempos na tabela estão convertidos em segundos. Logo multiplicamos os tempos dados pela método por $10^{(-9)}$.

Pela análise das médias e do desvio padrão de cada grafo, podemos constatar que o grafo1 foi o que menos se desviou da média ou seja o que teve menor variação de tempo em relação aos demais. Em seguida vem o grafo5, grafo3, grafo4 e finalmente o grafo2.

