Nome: Lukas Freitas de Carvalho

Matrícula: 202376033

Descrição do problema:

Coloração de arestas consiste em atribuir cores às arestas de um grafo de modo que arestas adjacentes (que compartilham um vértice comum) recebam cores diferentes. O objetivo principal é determinar o número cromático de arestas, ou seja, o menor número de cores necessário para colorir o grafo sem conflitos.

Descrição das instâncias:

Lista com os nomes dos arquivos das instâncias na pasta de entradas:

3elt.txt
power.txt
ca-GrQc.txt
EPA.txt
Erdos972.txt
G55.txt
inf-power.txt
G57.txt
socfb-Rochester38.txt
scc_rt_bahrain.txt

Todos as instâncias foram pegas na Network Repository;

Nome na Network Repository:

3elt

Grafo com 4720 vértices e 13722 arestas
MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

power

Grafo com 4941 vértices e 6594 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

ca-GrQc

Grafo com 4158 nós e 13422 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

EPA

Grafo com 4772 vértices e 8965 arestas

MatrixMarket matrix coordinate pattern general

Grafo direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

Erdos972

Grafo com 5488 vértices e 7085 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

G55

Grafo com exatamente 5 mil nós e 12498 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

inf-power

Grafo com 4941 vértices e 6594 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

G57

Grafo com 5000 vértices e 10000 arestas

MatrixMarket matrix coordinate integer symmetric

Grafo não direcionado, ponderado nas arestas

Nome na Network Repository:

socfb-Rochester38

Grafo com 4563 vértices e 161404 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Nome na Network Repository:

scc_rt_bahrain

Grafo com 4659 vértices e 129 arestas

MatrixMarket matrix coordinate pattern symmetric

Grafo não direcionado, não ponderado tanto em vértices quanto em arestas

Descrição dos métodos implementados

Coloração de arestas Guloso:

Percorri de forma crescente todas as arestas e, para cada aresta não colorida, eu percorria as arestas adjacentes verificando quais cores foram usadas, se uma cor foi usada, eu marco como usada e vou percorrendo até acabarem as arestas adjacentes, no fim, se uma cor estiver marcada como não usada, coloro a aresta com a primeira cor não usada disponível, se todas já tiverem sido usadas, adiciono uma nova cor.

Coloração de arestas Randomizado:

Seleciono aleatoriamente uma aresta no grafo que ainda não tenha sido colorida e, para cada aresta não colorida, eu verifico seus adjacentes para ver quais cores estão disponíveis e listá-las, por fim, seleciono dentre as cores disponíveis listadas aleatoriamente uma cor e marco a aresta com a cor.

Coloração de arestas Reativo:

Seleciono aleatoriamente uma aresta no grafo que ainda não tenha sido colorida e, para cada aresta não colorida, eu verifico seus adjacentes para ver quais cores estão disponíveis e listá-las, seleciono aleatoriamente entre uma coloração aleatória ou uma coloração gulosa e, para cada aresta colorida sem aumentar o número de cores atribuo uma recompensa para o método que foi escolhido, de forma que ele tenha mais chances de ser utilizado na próxima execução.

Análise de tempo de execução entre a lista e a matriz

Rodei o programa 3 vezes e as médias dos tempos de execução na matriz de adjacência foram:

3elt

Algoritmo guloso:

Cores retornadas: 10;

Na matriz de adjacência:
Real: 1.31 s
User: 1.07 s
Sys: 0.23 s

Na Lista Encadeada:
real 448.23

user 448.09
sys 0.10

Algoritmo Randomizado:

Cores retornadas: Entre 11 e 12

Na matriz de adjacência:
Real: 14.22 s
User: 14.05 s
Sys: 0.21 s

Na Lista Encadeada:
real 452.92
user 452.78
sys 0.12

Algoritmo Reativo:

Cores retornadas: Entre 11 e 12

Na matriz de Adjacência
Real: 12.46 s
User: 12.24 s
Sys: 0.21 s

Na Lista Encadeada:
real 452.43
user 452.26
sys 0.11

power

Algoritmo Guloso:

Cores retornadas: 19

Na matriz de Adjacência
Real 11.89
User 11.70
Sys 0.18

Na Lista Encadeada:
real 521.68
user 521.53
sys 0.11

Algortimo Randomizado:

Cores retornadas: 19

Na matriz de Adjacência
Real 11.01
User 10.80
Sys 0.19

Na Lista Encadeada:
real 533.22
user 533.02
sys 0.10

Algoritmo Reativo:

Cores retornadas: 19

Na matriz de Adjacência
Real 10.75
User 10.56
Sys 0.19

Na Lista Encadeada:
real 540.20
user 540.00
sys 0.12

EPA

Algoritmo Guloso:

Cores retornadas: 175

Na matriz de Adjacência
real 1.45
user 1.20
sys 0.24

Na Lista Encadeada:
real 460.69
user 460.53
sys 0.11

Algortimo Randomizado:

Cores retornadas: 175

Na matriz de Adjacência
real 10.53
user 10.30
sys 0.22

Na Lista Encadeada:
real 462.28 s
user 462.13 s
sys 0.11 s

Algoritmo Reativo:

Cores retornadas: 175

Na matriz de Adjacência
real 12.51
user 12.29
sys 0.22

Na Lista Encadeada:
real 469.82
user 469.70
sys 0.11

Erdos972

Algoritmo Guloso:

Cores retornadas: 61

Na matriz de Adjacência
real 1.88
user 1.32
sys 0.55

Na Lista Encadeada:
real 739.82
user 739.60
sys 0.15

Algortimo Randomizado:

Cores retornadas: 61

Na matriz de Adjacência
real 16.57

user 16.02
sys 0.54

Na Lista Encadeada:
real 750.73
user 750.50
sys 0.14

Algoritmo Reativo:
Cores retornadas: 61

Na matriz de Adjacência
real 15.91
user 15.35
sys 0.56

Na Lista Encadeada:
real 752.16
user 751.79
sys 0.16

G55

Algoritmo Guloso:
Cores retornadas: 15

Na matriz de Adjacência
real 1.48
user 1.20
sys 0.27

Na Lista Encadeada:
real 530.37

user 530.23
sys 0.13

Algortimo Randomizado:

Cores retornadas: Entre 15 e 16

Na matriz de Adjacência
real 11.46
user 11.23
sys 0.23

Na Lista Encadeada:
real 533.00
user 532.84
sys 0.12

Algoritmo Reativo:

Cores retornadas: Entre 15 e 16

Na matriz de Adjacência
real 13.71
user 13.47
sys 0.23

Na Lista Encadeada:
real 534.89
user 534.71
sys 0.13

inf-power

Algoritmo Guloso:

Cores retornadas: 19

Na matriz de Adjacência
real 1.28
user 1.01
sys 0.27

Na Lista Encadeada:
real 531.04
user 530.89
sys 0.11

Algortimo Randomizado:

Cores retornadas: 19

Na matriz de Adjacência
real 12.65
user 12.43
sys 0.21

Na Lista Encadeada:
real 527.01
user 526.84
sys 0.11

Algoritmo Reativo:

Cores retornadas: 19

Na matriz de Adjacência
real 13.61
user 13.37
sys 0.22

Na Lista Encadeada:

real 515.14
user 514.98
sys 0.12

G57

Algoritmo Guloso:

Cores retornadas: 7

Na matriz de Adjacência
real 13.25
user 13.06
sys 0.19

Na Lista Encadeada:

Algortimo Randomizado:

Cores retornadas: 7

Na matriz de Adjacência
real 12.78
user 12.58
sys 0.19

Na Lista Encadeada:
real 561.94
user 561.78
sys 0.12

Algoritmo Reativo:

Cores retornadas: 7

Na matriz de Adjacência
real 12.01
user 11.81
sys 0.19

Na Lista Encadeada:
real 567.26
user 567.12
sys 0.12

scc_rt_bahrain

Algoritmo Guloso:

Cores retornadas: 20

Na matriz de Adjacência
real 0.94
user 0.71
sys 0.23

Na Lista Encadeada:
real 452.55
user 452.42
sys 0.10

Algortimo Randomizado:

Cores retornadas: 20

Na matriz de Adjacência
real 5.34
user 5.13
sys 0.20

Na Lista Encadeada:
real 454.47
user 454.28
sys 0.11

Algoritmo Reativo:

Cores retornadas: 20

Na matriz de Adjacência
real 7.60
user 7.37
sys 0.22

Na Lista Encadeada:
real 454.36
user 454.26
sys 0.10

socfb-Rochester38

Algoritmo Guloso:

Cores retornadas: 1224

Na matriz de Adjacência
real 5.27
user 5.06
sys 0.20

Na Lista Encadeada:
real 452.11
user 451.97
sys 0.11

Algortimo Randomizado:

Cores retornadas: 1224

Na matriz de Adjacência
real 18.45
user 18.24
sys 0.20

Na Lista Encadeada:
real 464.56
user 464.42
sys 0.12

Algoritmo Reativo:

Cores retornadas: 1224

Na matriz de Adjacência
real 21.48
user 21.25
sys 0.22

Na Lista Encadeada:
real 466.46
user 466.34
sys 0.11

ca-GrQc

Algoritmo Guloso:

Cores retornadas: 81

Na matriz de Adjacência

real 1.05
user 0.85
sys 0.19

Na Lista Encadeada:
real 325.31
user 325.20
sys 0.08

Algortimo Randomizado:

Cores retornadas: 81

Na matriz de Adjacência
real 13.93
user 13.74
sys 0.19

Na Lista Encadeada:
real 333.20
user 333.09
sys 0.09

Algoritmo Reativo:

Cores retornadas: 81

Na matriz de Adjacência
real 10.09
user 9.91
sys 0.17

Na Lista Encadeada:
real 334.55
user 334.41

sys 0.09

Análise de tempo de resultado com teste de hipótese entre os métodos:

No geral, a matriz foi mais de 300 vezes mais rápida do que a lista encadeada. Isso ocorre devido a forma como implementei a lista encadeada, os métodos e a própria velocidade de busca da lista encadeada.
A matriz com complexidade de acesso O(1) para as arestas e a lista encadeada com complexidade de $O(N^2)$ atrasam e muito a lista para resolução desses problemas, além disso, para minimizar as diferenças, eu criei uma matriz em cada método com o intuito de diminuir a quantidade de requisições das arestas, já que inicialmente a lista encadeada era cerca de 1000x mais lenta, levando quase cerca de 30 minutos para resolver cada instância.
Além disso, o método guloso apresentou os melhores resultados de tempo, já que ele só percorre cada aresta uma única vez, enquanto nos métodos randomizado e reativo ele pode solicitar mais de uma vez uma aresta já colorida, o que aumenta na maioria dos casos o tempo de execução do programa.

Conclusões:

As conclusões que posso tirar com base nesses resultados é que a maioria dos algoritmos são de classe 1, apresentando cores de forma consistente, nessa classificação se enquadram os grafos bipartidos, grafos planares que não possuem cruzamentos ou grafos regulares de grau par.
Além disso, a matriz de adjacência se demonstrou muito mais eficiente do que a lista encadeada, tanto devido sua implementação quanto pela sua complexidade de acesso.