CENTRO UNIVERSITÁRIO
FEI MESTRADO ENGENHARIA ELÉTRICA
ALGORITMOS COMPUTACIONAIS – PEL201

RELATÓRIO 2
ALGORITMOS DE GRAFOS
PRIM E DIJKSTRA

GUSTAVO APARECIDO DE SOUZA VIANA – 120110-2
SÃO BERNARDO DO CAMPO
2020

O código fonte está disponível em:
https://github.com/apparecidoo/master-computational-algorithms

1)

O algoritmo Prim é um algoritmo guloso que tem como objetivo achar um árvore geradora mínima afim de achar o menor custo para interligar todos os vértices de um grafo sem haver repetição de ligações entre os vértices, consequentemente que minimize o peso total. Importante ressaltar que o grafo deverá ser não direcionado e conexo.

O raciocínio do algoritmo é percorrer cada vértice atualizando o seu peso baseado na menor aresta conectada com seus vizinhos. A cada iteração o vértice atual é atualizado com o menor peso das arestas conectadas aos vizinhos e quando isso é feito o vértice é marcado como finalizado (explorado) adicionando a solução final, daí que vem a característica de ser um algoritmo guloso pois a cada iteração é uma parte resolvida da árvore final, e assim sucessivamente até que todos os vértices sejam explorados.

2)

O algoritmo Djikstra é um algoritmo com o objetivo de achar uma árvore com o onde os pesos dos vértices são o menor caminho do mesmo até o vértice inicial sem haver repetições de conexões. Dijkstra não possui restrições com relação a ser direcionado ou não, mas os pesos das arestas deve ser positivas ou zero, ou seja, não negativa.

O raciocínio do Dijkstra é parecido com o do Prim, possui a mesma forma de iteração e exploração dos vértices calculando os pesos baseados nas arestas vizinhas mas a forma que é calculado o peso e atribuído ao vértice é diferente. O cálculo é baseado no vértice inicial, ou seja, a distância do vértice em questão até o inicial (soma do peso atribuído ao vértice conectado a ele mais a aresta que o interliga) acarretando em indiretamente em uma somatória, obviamente a menor distância.

3)

Os dois algoritmos apresentam uma árvore geradora onde há apenas uma conexão entre os vértices, mas com objetivos diferentes.

A primeira diferença é que o Dijkstra possui a flexibilidade de ser direcionado ou não, já o Prim necessariamente necessita ser não direcionado.

O Dijkstra tem objetivo de achar o menor caminho entre dois vértices, já o Prim acha o menor peso entre os vértices para que a arvore geradora tenha um custo mínimo na ligação de todos os vértices.

No Prim o cálculo do peso do vértice é baseado somente na menor aresta que o conecta, já no Dijkstra na menor soma do Vértice vizinho + Aresta que o conecta.

O Dijkstra é geralmente utilizado para descobrir a menor distância de um vértice a outro, e o Prim para descobrir o menor custo afim de visitar todos os vértices.

4)

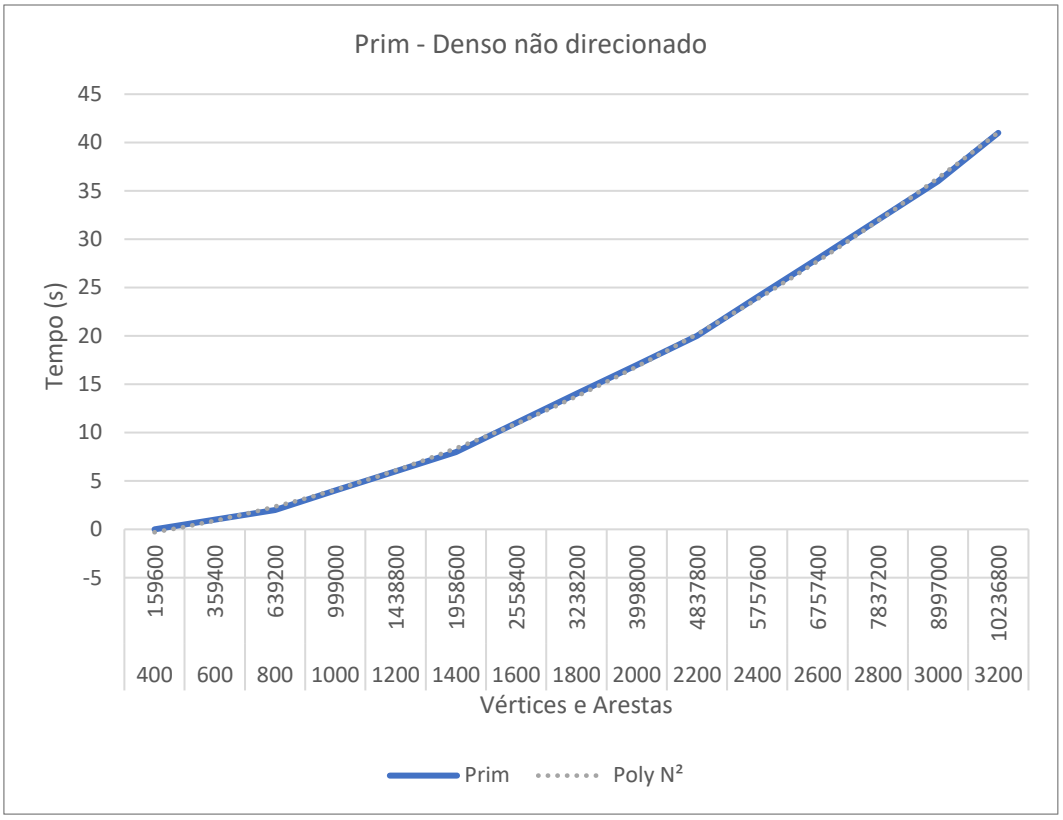
Grafo, onde existem funções para criar a matriz de adjacência:
https://github.com/apparecidoo/master-computational-algorithms/blob/master/ComputationalAlgorithms/graph.cpp
Prim, funções para aplicar o algoritmo Prim:
https://github.com/apparecidoo/master-computational-algorithms/blob/master/ComputationalAlgorithms/prim.h
Dijkstra, funções para aplicar o algoritmo Dijkstra:
https://github.com/apparecidoo/master-computational-algorithms/blob/master/ComputationalAlgorithms/djikstra.h

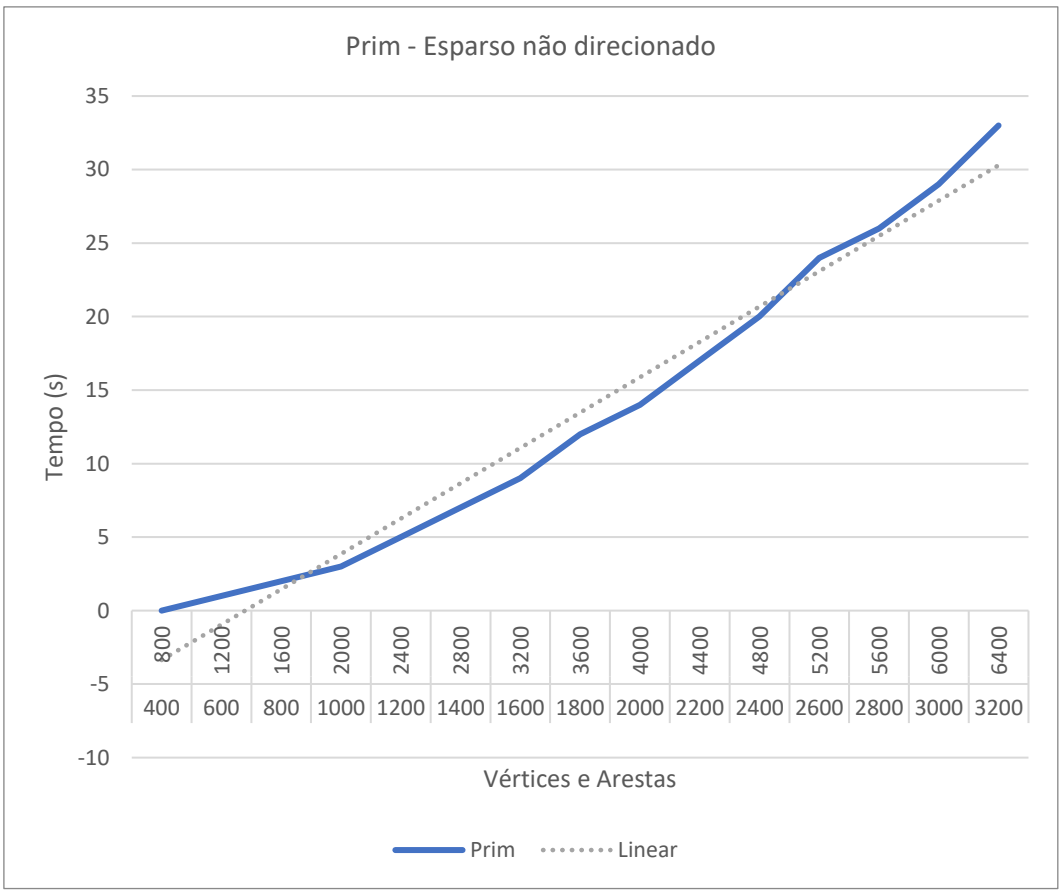
5)

Para os experimentos de desempenho foram gerados 15 grafos de 400 a 2200 vértices, para cada grafo foi executado o algoritmo Prim e Dijkstra 500 vezes. A seguir a tabela que mostra o tempo de execução em microssegundos para cada algoritmo e o gráfico correspondente.

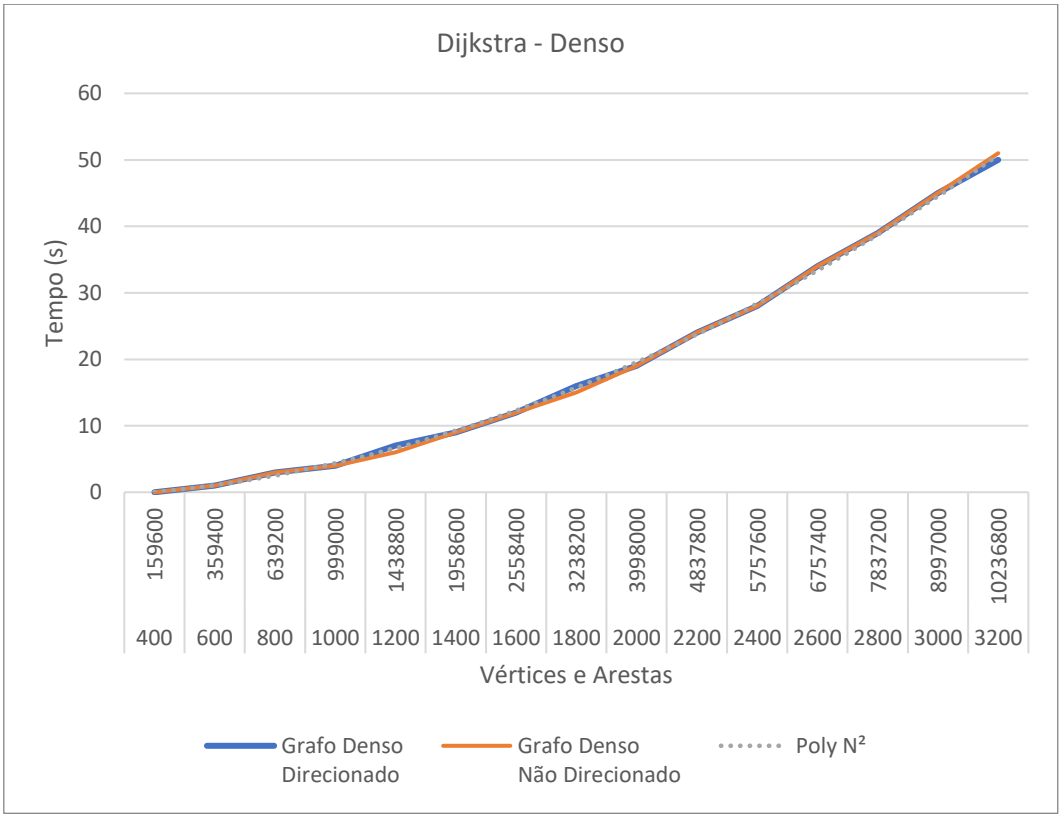
Podemos concluir a partir do gráfico que o algoritmo Prim e Dijkstra seguem a complexidade N^2 no pior caso, pois no grafo denso de ambos apresentou uma linha de função tendendo a N^2. No caso esparso apresentou ser linear a complexidade.

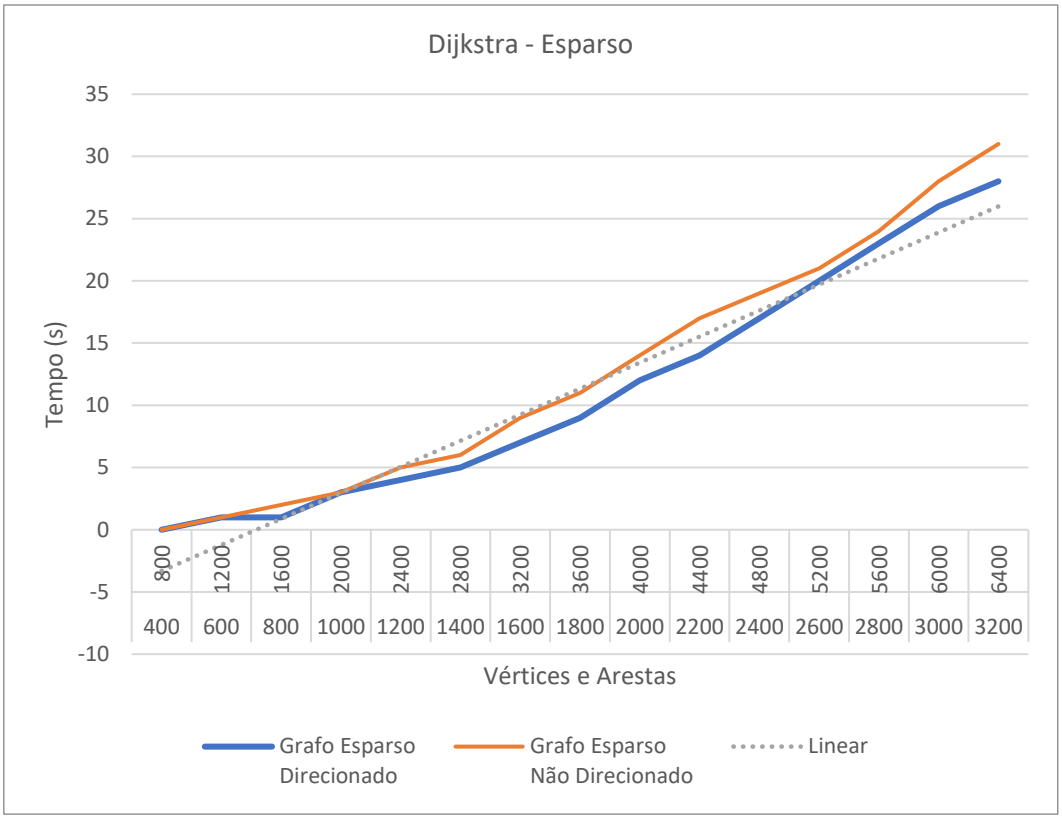
Algoritmo Prim			Grafo Denso Não Direcionado		Grafo Esparso Não Direcionado	
Iteração	Repetição	Vértices	Arestas	Tempos (s)	Arestas	Tempo (s)
1	500	400	159600	0	1594	0
2	500	600	359400	1	2394	1
3	500	800	639200	2	3194	2
4	500	1000	999000	4	3994	3
5	500	1200	1438800	6	4794	5
6	500	1400	1958600	8	5594	7
7	500	1600	2558400	11	6394	9
8	500	1800	3238200	14	7194	12
9	500	2000	3998000	17	7994	14
10	500	2200	4837800	20	8794	17
11	500	2400	5757600	24	9594	20
12	500	2600	6757400	28	10394	24
13	500	2800	7837200	32	11194	26
14	500	3000	8997000	36	11994	29
15	500	3200	10236800	41	12794	33


Prim - Denso não direcionado
Tempo (s)
45
40
35
30
25
20
15
10
5
0
-5
159600
359400
639200
999000
1438800
1958600
2558400
3238200
3998000
4837800
5757600
6757400
7837200
8997000
10236800
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
3000
3200
Vértices e Arestas
Prim
Poly N²


Prim - Esparso não direcionado
Tempo (s)
35
30
25
20
15
10
5
0
-5
-10
800
1200
1600
2000
2400
2800
3200
3600
4000
4400
4800
5200
5600
6000
6400
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
3000
3200
Vértices e Arestas
Prim
Linear

Algoritmo Dijkstra			Grafo Denso Direcionado		Grafo Esparso Direcionado		Grafo Denso Não Direcionado		Grafo Esparso Não Direcionado	
Iteração	Repetição	Vértices	Arestas	Tempos (s)	Arestas	Tempos (s)	Arestas	Tempos (s)	Arestas	Tempos (s)
1	500	400	159600	0	800	0	159600	0	800	0
2	500	600	359400	1	1200	1	359400	1	1200	1
3	500	800	639200	3	1600	1	639200	3	1600	2
4	500	1000	999000	4	2000	3	999000	4	2000	3
5	500	1200	1438800	7	2400	4	1438800	6	2400	5
6	500	1400	1958600	9	2800	5	1958600	9	2800	6
7	500	1600	2558400	12	3200	7	2558400	12	3200	9
8	500	1800	3238200	16	3600	9	3238200	15	3600	11
9	500	2000	3998000	19	4000	12	3998000	19	4000	14
10	500	2200	4837800	24	4400	14	4837800	24	4400	17
11	500	2400	5757600	28	4800	17	5757600	28	4800	19
12	500	2600	6757400	34	5200	20	6757400	34	5200	21
13	500	2800	7837200	39	5600	23	7837200	39	5600	24
14	500	3000	8997000	45	6000	26	8997000	45	6000	28
15	500	3200	10236800	50	6400	28	10236800	51	6400	31




Dijkstra - Esparso
35
30
25
20
15
10
5
0
-5
-10
Tempo (s)
800
1200
1600
2000
2400
2800
3200
3600
4000
4400
4800
5200
5600
6000
6400
400
600
800
1000
1200
1400
1600
1800
2000
2200
2400
2600
2800
3000
3200
Vértices e Arestas
Grafo Esparso Direcionado
Grafo Esparso Não Direcionado
Linear