

## Introdução

Grafos são estruturas que permitem modelar problemas complexos, representando relações entre objetos de forma eficiente. Neste experimento, foram implementados e testados algoritmos para grafos, com o objetivo de medir os desempenho com diferentes entradas.

## Objetivo

Implementar e analisar o desempenho de algoritmos de grafos em ambas representações (lista de adjacência ou matriz de adjacência), algoritmos como busca em largura, busca em profundidade, ordenação topológica, árvore geradora mínima (algoritmos de Kruskal e de Prim), caminho mínimo de origem única para todos vértices (algoritmos de Bellman-Ford e de Dijkstra), caminho mínimo entre todos pares de vértices (algoritmo de Floyd-Warshall) e fluxo máximo em redes (algoritmo de Ford-Fulkerson), utilizando diferentes entradas e identificando fatores que influenciam na eficiência dos algoritmos. Os algoritmos de Bellman-Ford e de Dijkstra, devem ser executados 10 vezes (para diferentes amostras), para responder se o desempenho é semelhante quando utilizado um grafo denso ou esparso, por lista de adjacência ou matriz de adjacência.

## Metodologia

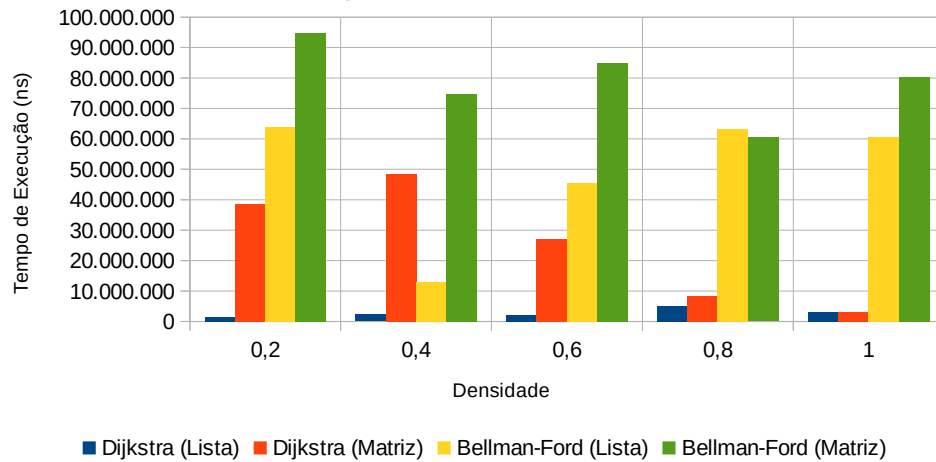
Os algoritmos propostos no objetivo foram devidamente implementados utilizando a internet e os pseudo-algoritmos fornecidos pelo docente. Os algoritmos de Bellman-Ford e de Dijkstra foram executados 10 vezes para cada uma das 15 amostras fornecidas pelo docente, que variavam a densidade, sendo cinco amostras para grafos com 100, 200 e 500 vértices. Os tempos de execução foram registrados e utilizados para a criação de tabelas e gráficos.

## Resultados

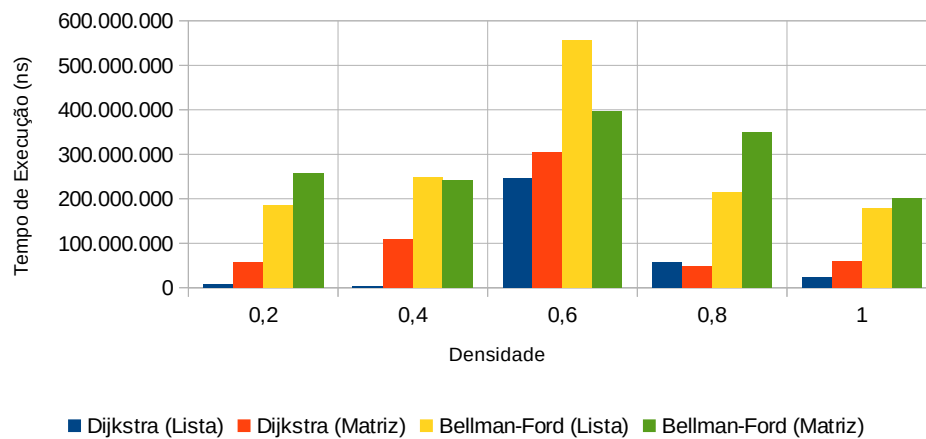
Com os tempos médios em nanosegundos, foi criada uma tabela com os resultados. A partir dela, foram gerados três gráficos, variando o número de vértices, para facilitar a análise do desempenho dos algoritmos com base na representação e na densidade dos grafos.

Vértices	Densidade	Dijkstra (Lista)	Dijkstra (Matriz)	Bellman-Ford (Lista)	Bellman-Ford (Matriz)
100	0,2	1.338.700	38.623.190	63.771.530	94.816.410
	0,4	2.288.510	48.552.130	13.016.320	74.744.730
	0,6	2.101.210	27.204.570	45.448.650	84.917.990
	0,8	5.247.800	8.246.900	63.189.780	60.436.720
	1	2.970.760	3.048.060	60.672.000	80.221.900
200	0,2	7.898.320	57.280.420	184.666.870	257.413.910
	0,4	3.194.570	108.039.740	247.726.670	240.971.300
	0,6	246.386.080	304.249.820	556.316.800	397.103.050
	0,8	56.515.580	47.318.720	215.579.450	348.869.330
	1	24.104.340	58.470.300	178.168.880	202.108.040
500	0,2	10.848.980	88.972.800	462.981.450	153.908.210
	0,4	59.205.240	79.672.880	1.322.642.600	2.187.798.600
	0,6	58.709.620	92.190.610	3.204.767.600	8.848.483.200
	0,8	73.664.670	72.526.260	6.276.640.400	10.557.635.100
	1	90.241.930	59.245.340	9.548.297.480	10.813.434.360

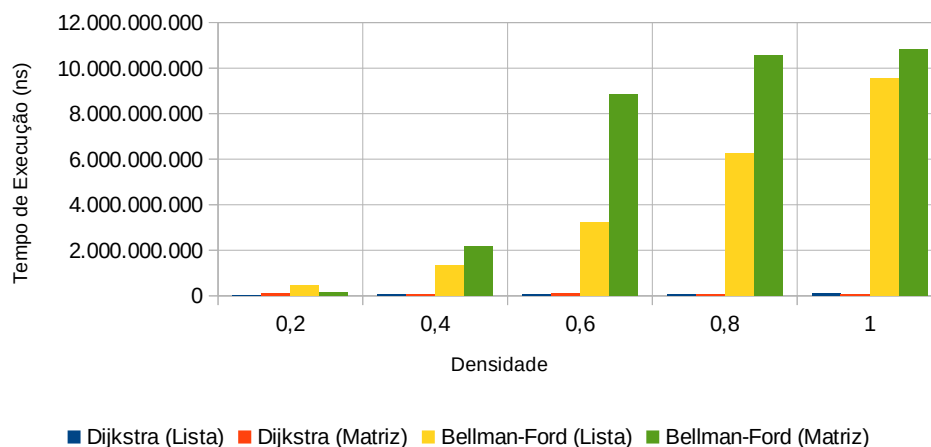
Desempenho dos Algoritmos Dijkstra e Bellman-Ford por Representação e Densidade com 100 Vértices



Desempenho dos Algoritmos Dijkstra e Bellman-Ford por Representação e Densidade com 200 Vértices



Desempenho dos Algoritmos Dijkstra e Bellman-Ford por Representação e Densidade com 500 Vértices



Analisando os gráficos, evidenciou-se que o algoritmo de Dijkstra é mais eficiente que o de Bellman-Ford. Além disso, para responder à questão sobre o desempenho dos tipos (denso/esparso) e representações (lista/matriz) de grafos serem semelhantes, constatou-se que não são. A lista é mais eficiente em grafos esparsos e tende a ser mais eficiente que a matriz, mas a diferença entre as representações diminui à medida que a densidade aumenta.

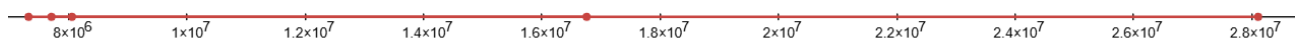
Ademais, os gráficos também destacaram anomalias (outliers) nos tempos de execução. Isso deve-se por conta das 10 execuções do algoritmo Bellman-Ford (representação por lista) com grafos de 100 vértices e densidade 0,4 (sample100-3960), e grafos de 200 vértices e densidade 0,6 (sample200-23880). Para analisar essas execuções específicas, foi criada uma tabela adicional e gerados gráficos de box plot.

Execução 1	Execução 2	Execução 3	Execução 4	Execução 5
188.398.000	352.393.700	59.194.700	113.462.500	76.573.200
27.113.300	28.104.800	19.155.600	8.077.000	8.026.600

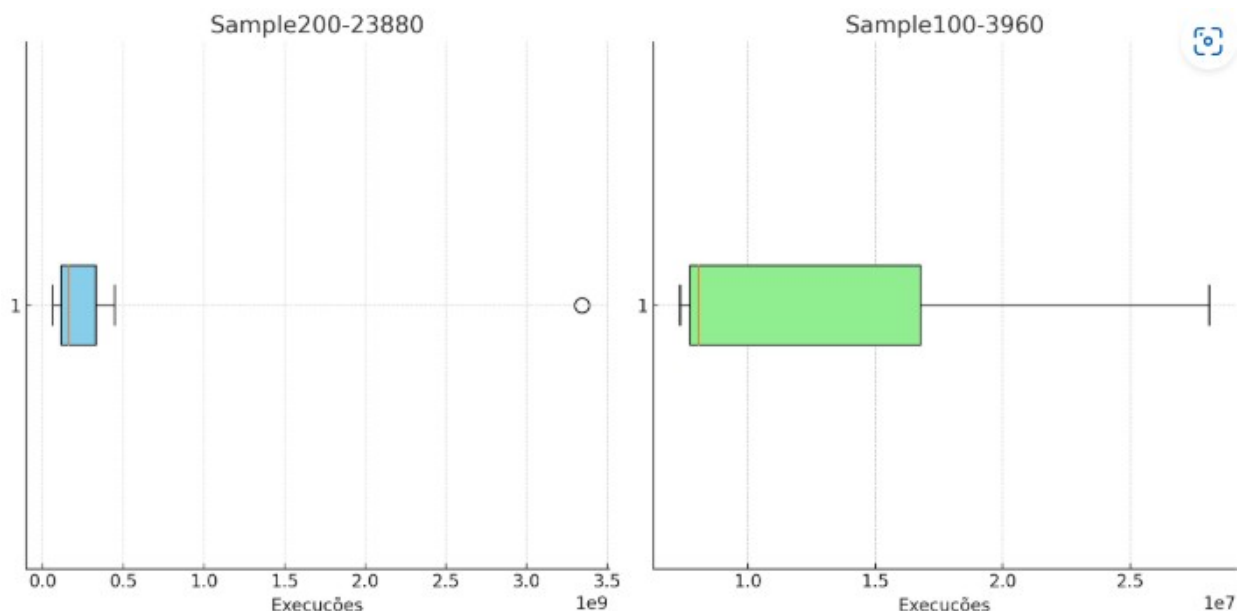
  

Execução 6	Execução 7	Execução 8	Execução 9	Execução 10	Vértices
126.643.300	3.344.198.900	258.674.300	114.614.900	445.807.500	200
7.687.400	7.318.500	9.548.500	7.378.300	7.753.200	100

Sample100-3960



sample200-23880



Os resultados mostraram que a amostra sample200-23880 apresentou maior dispersão nos tempos de execução (como o valor 3344198900), enquanto a sample100-3960 foi mais consistente, com tempos mais homogêneos, embora alguns ainda estivessem fora do esperado.

### **Considerações Finais**

Os experimentos realizados evidenciaram diferenças significativas entre os algoritmos e os tipo e representações de grafos. O algoritmo de Dijkstra demonstrou maior eficiência em todos os cenários. Quanto às representações, a lista se mostrou mais eficiente em grafos esparsos. Entretanto, a diferença entre lista e matriz diminuiu à medida que a densidade dos grafos aumentava, indicando que a escolha da representação deve considerar a estrutura e as características dos grafos analisados. Contudo, o estudo foi de grande importância para um melhor entendimento dos algoritmos utilizando, possibilitando maior entendimento à respeito de grafos.