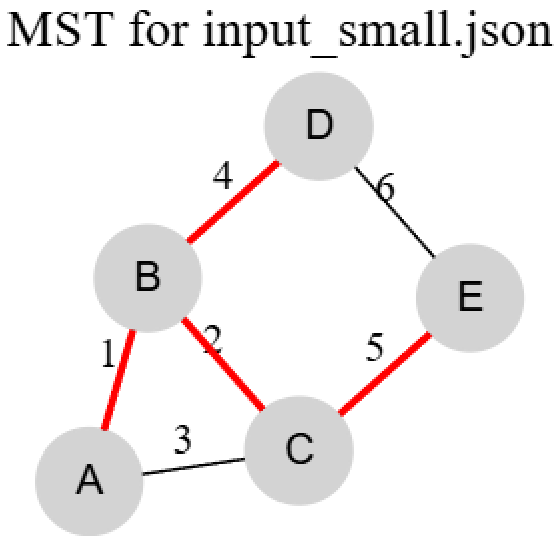
Aibyn Samat SE – 2438

1.

Dataset	Vertices	Edges	Prim Total Cost	Kruskal Total Cost	Prim Ops	Kruskal Ops	Prim Time (ms)	Kruskal Time (ms)
Small	5	6	12.0	12.0	5	5	3.19	0.70
Medium	10	13	26.0	26.0	10	10	3.57	0.72
Large	20	30	69.0	69.0	19	19	3.69	1

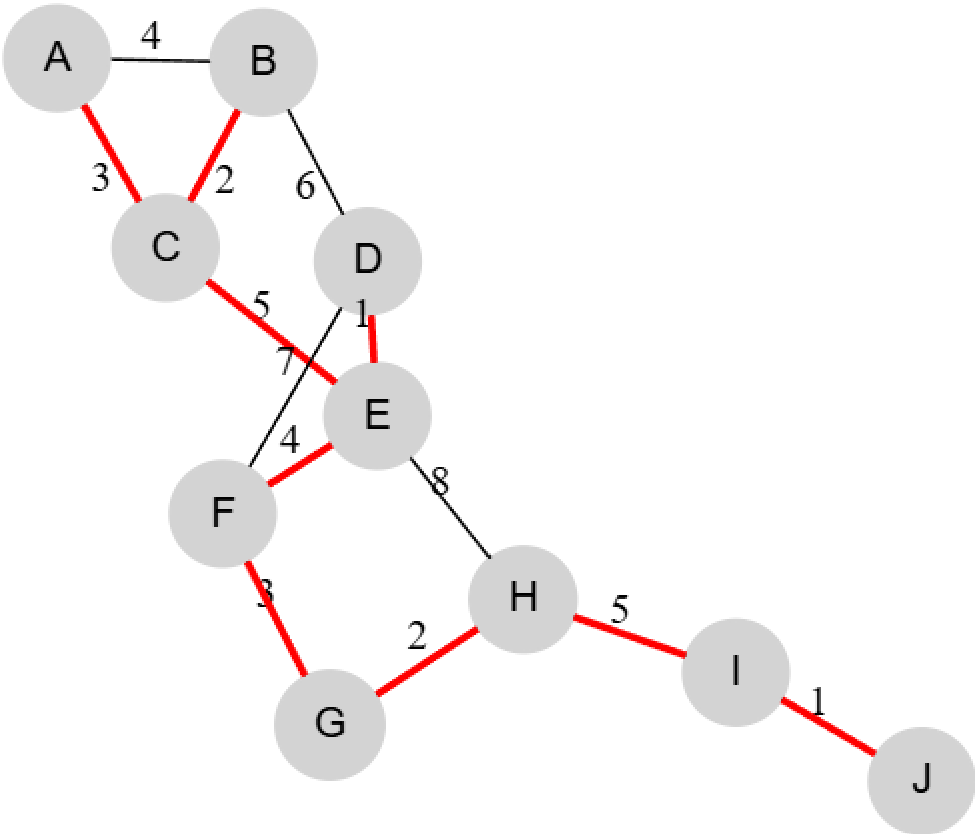
2.

1) Small graph with 5 vertices



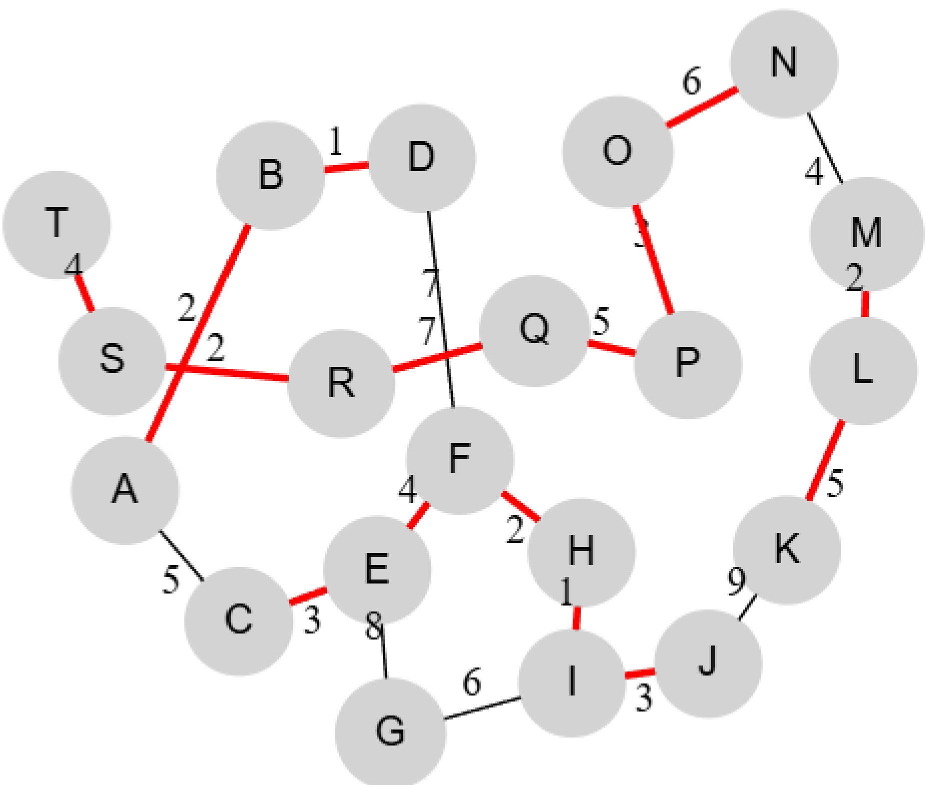
2) Medium graph with 10 vertices

Minimum Spanning Tree — Medium Graph

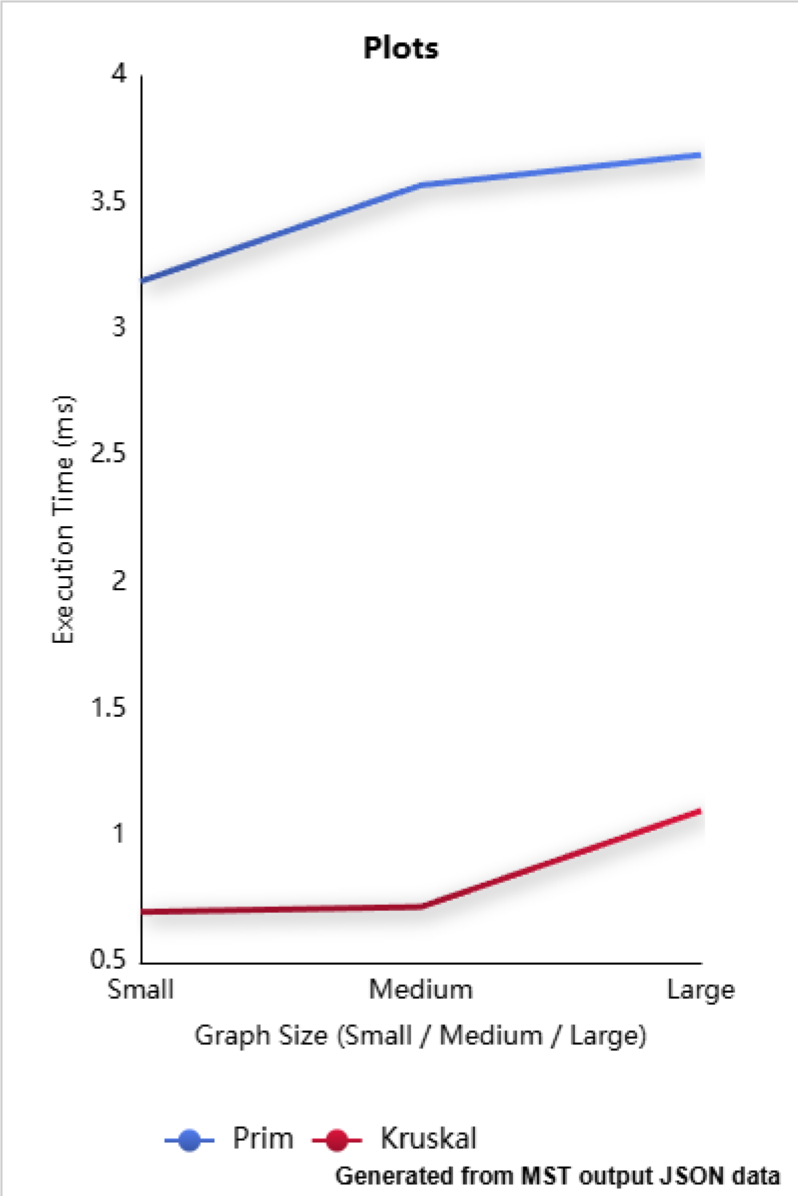


3) Large graph with 19 vertices

Minimum Spanning Tree — Large Graph



3.



Both algorithms have the same asymptotic complexity but differ in constants and data structure usage. Kruskal's algorithm iterates over sorted edges and uses Union-Find, making it faster on sparse graphs. Prim's algorithm grows vertex-by-vertex with a heap and performs better on dense graphs.

4.

Conclusion

Both Prim's and Kruskal's algorithms produced identical MST costs, confirming correctness. Kruskal's algorithm performs faster and with fewer operations across datasets. Prim's algorithm is preferable for dense graphs, while Kruskal's excels in sparse or edge-based representations.

5.

References

1) Graph visualizations generated via Graphviz Online (https://dreampuf.github.io/GraphvizOnline/)
2) Plots via ChartGo (https://www.chartgo.com/get.do?id=cc92faf0b6)
3) Sedgewick, R. & Wayne, K. *Algorithms (4th Edition)* — Section 4.3 Minimum Spanning Trees