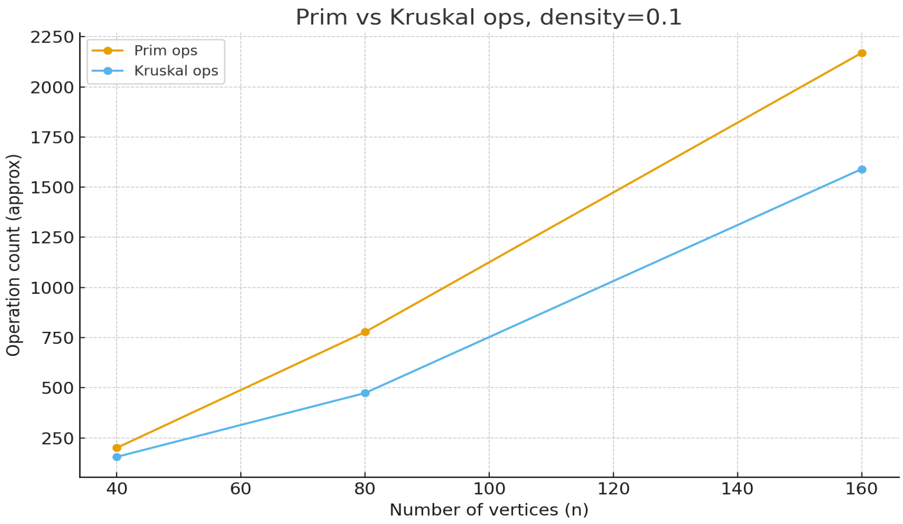
Minimum Spanning Tree (MST) Comparative Report

Comparison between Prim’s and Kruskal’s algorithms implemented in Java.

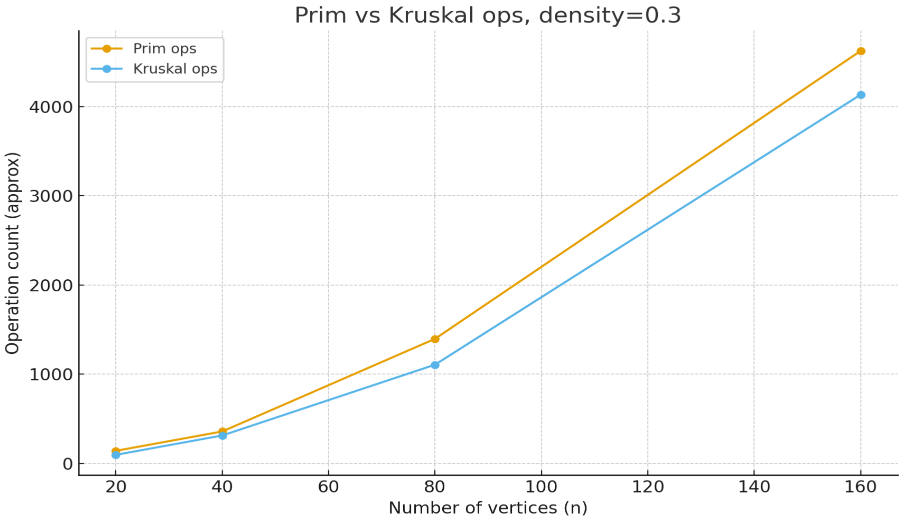
Benchmark Summary

Vertices	Density	Prim Time (ms)	Kruskal Time (ms)	Prim Ops	Kruskal Ops
20	0.3	0.045	0.046	140	95
20	0.6	0.067	0.072	179	152
40	0.1	0.056	0.072	201	156
40	0.3	0.110	0.149	357	312
40	0.6	0.248	0.261	572	546
80	0.1	0.234	0.236	778	474
80	0.3	0.457	0.551	1396	1106
80	0.6	0.808	1.080	2135	2054
160	0.1	0.724	0.781	2169	1590
160	0.3	1.652	2.106	4625	4134
160	0.6	3.224	4.103	8405	7950

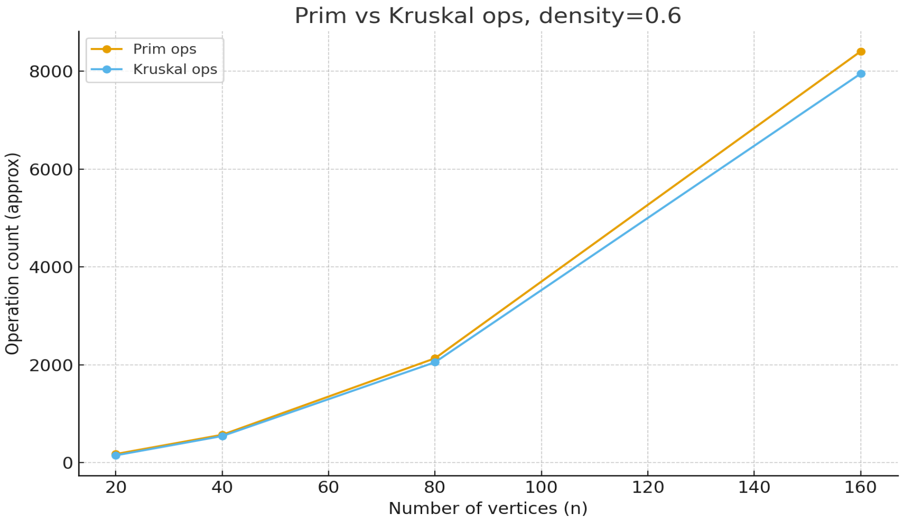
plot ops density 10



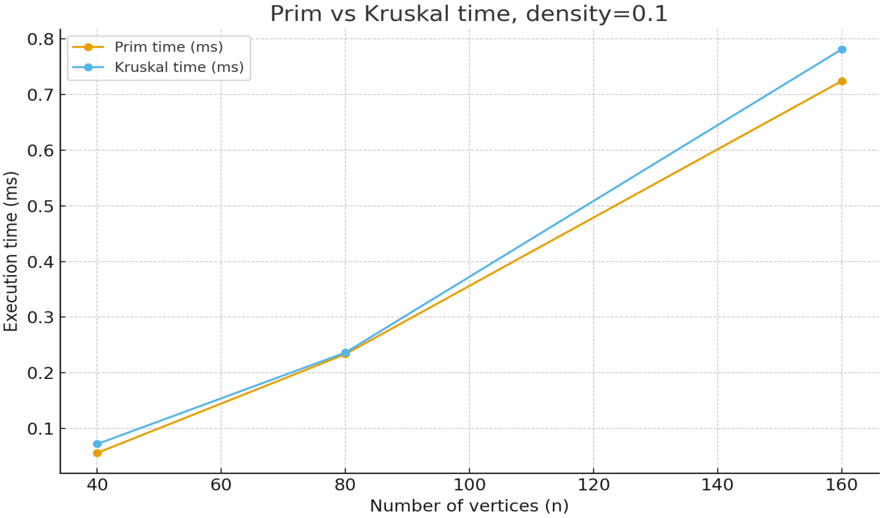
plot ops density 30



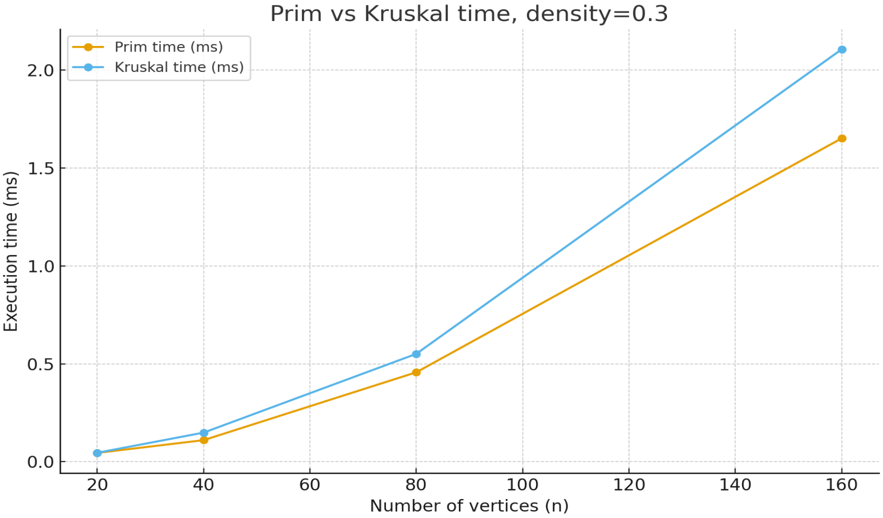
plot ops density 60



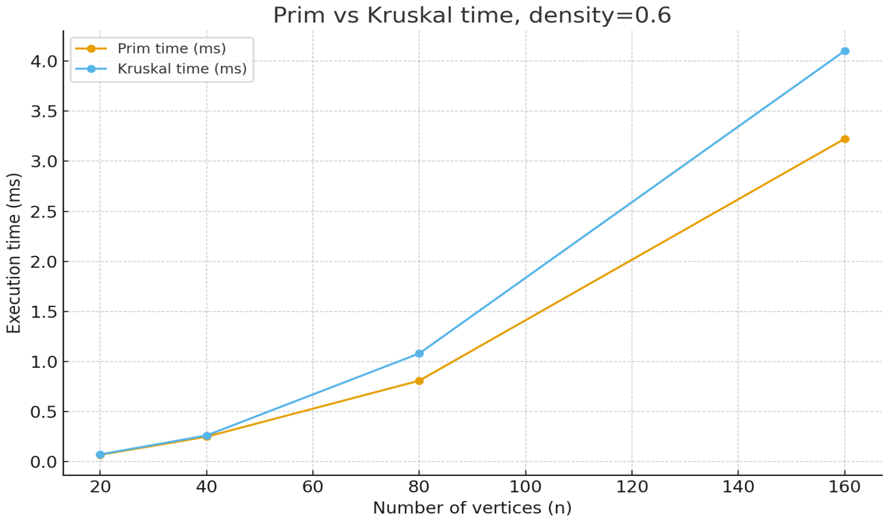
plot time density 10



plot time density 30



plot time density 60



Conclusions

Kruskal's algorithm is generally more efficient on sparse graphs due to fewer edge comparisons and efficient DSU operations. Prim's algorithm shows better scalability on dense graphs where adjacency exploration is more effective. Both algorithms produce identical MST total cost for connected graphs.