|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Graph Type** | **Nodes** | **Edges** | **Execution Time (s)** | **Energy (J)** | **Peak Memory (KB)** | **CO₂ Emissions (kg, BD)** |
| Prims | Sparse | 50 | 125 | 0.534212 | 34.72 | 1500 | 0.000006 |
| Kruskal | 50 | 125 | 0.789654 | 51.32 | 1800 | 0.000009 |
| Prims | Dense | 50 | 1225 | 3.214567 | 208.95 | 2500 | 0.000036 |
| Kruskal | 50 | 1225 | 5.876543 | 381.97 | 3200 | 0.000066 |

**Discussion / Conclusion**

**Sparse Graph:**

We know theoretically that Prim’s Algorithm has **O(E log V)** time complexity, while Kruskal’s Algorithm has **O(E log E)**. Putting **V = 50** and **E = 125** gives Prim’s 705 and Kruskal 871; the output also matches our results. Although the result can vary, Kruskal can be faster for small graphs because of the repeated edges pushing in the heap for Prim’s, resulting in heavier overhead.

**Dense Graph:**

Putting the values V = 50 and calculating E = 1225 for a dense graph gives Prim’s 6910 and Kruskal 12,568, which clearly matches our output. Although the time complexity can be even better for Prim’s in dense graphs using the adjacency matrix with linear searching or a Fibonacci heap, resulting in O(V²).

**Overall:** For carbon emission, Prim’s is better.