

Algorithm	Graph Type	Nodes	Edges	Execution Time (s)	Energy (J)	Peak Memory (KB)	CO ₂ Emissions (kg, BD)
Johnson	Sparse	50	125	0.0017344	0.112736	5812224	1.94156e-008
Floyd-Warshall		50	125	0.0015113	0.0982345	5881856	1.69182e-008
Johnson	Dense	50	1225	0.0105505	0.685783	5918720	1.18107e-007
Floyd-Warshall		50	1225	0.0060698	0.394537	5918720	6.7948e-008

Sparse Graph:

We know the time complexity of Johnson's algorithm is $O(VE \log V)$, while Floyd-Warshall is $O(V^3)$. A sparse graph is an incomplete graph where not all nodes are connected. From the time complexity, we might assume that Johnson should be faster. However, in our practical case, Johnson is slightly slower due to overhead issues, such as push/pop operations in the priority queue, creating vectors, and converting the edge list to an adjacency list for Dijkstra. These extra steps are required by the program but are not part of the core algorithm. On the other hand, Floyd-Warshall uses triple nested loops with no such overhead. Therefore, for a small graph with 50 nodes, Floyd-Warshall performs better. However, as the number of nodes increases, Johnson outshines Floyd-Warshall, making it better for large sparse graphs.

Dense Graph:

In a dense graph, the number of edges $E \approx V^2$. Considering the time complexities, Floyd-Warshall remains $O(V^3)$, while Johnson becomes $O(V^3 \log V)$. This clearly indicates that Floyd-Warshall is more efficient than Johnson for dense graphs.