Kruskal’s algorithm and Prim’s algorithm are both used to find the minimum spanning tree of a weighted graph. Here are the differences between the two algorithms:

Algorithm	Approach	Data Structure	Time Complexity
Kruskal’s Algorithm	Greedy	Disjoint Set	O(E log E)
Prim’s Algorithm	Greedy	Priority Queue	O(E log V)

1. Approach: Kruskal’s algorithm is a Greedy approach that starts with an empty spanning tree and adds edges to it one at a time, while ensuring that no cycle is formed. Prim’s algorithm, on the other hand, starts with a single vertex and adds edges to it until it forms a spanning tree.

2. Edge selection: Kruskal’s algorithm selects the minimum weighted edge that does not form a cycle at each step, while Prim’s algorithm selects the minimum weighted edge that connects a vertex in the tree to a vertex outside the tree.

3. Data structure: Kruskal’s algorithm uses a Disjoint Set Data Structure to maintain the disjoint sets of vertices and detect cycles. Prim’s algorithm uses a priority queue or a heap to select the minimum weighted edge.

4. Running time: The worst-case running time of Kruskal’s algorithm is O(E log E), where E is the number of edges in the graph. The worst-case running time of Prim’s algorithm is O(E log V), where V is the number of vertices in the graph.

5. Complexity: Kruskal’s algorithm has a space complexity of O(E), while Prim’s algorithm has a space complexity of O(V).

6. Performance: Kruskal’s algorithm performs better on sparse graphs with a large number of edges, while Prim’s algorithm performs better on dense graphs with a large number of vertices.