Dijkstra's algorithm, named after Dutch computer scientist, Edsger W. Dijkstra, is an algorithm designed to find the shortest path between any two nodes in a weighted graph. The algorithm works by maintaining a priority queue of unexplored nodes, and repeatedly selecting the shortest path from the source node to that node. The algorithm then visits any and all neighbors of the selected node, updating their distances if necessary and adding them to the priority queue. The algorithm repeats until the destination node is selected or the priority queue is empty.

The algorithm is implemented by first assigning distance values to every node relationship, this is easiest to visualize when you first set the source node to zero and infinity for all other nodes. Second you add the source node to the priority queue, check that the priority queue is not empty, and while it isn't, do the following:

1. Remove the node with the smallest tentative distance from the priority queue.
2. For each neighbor of the current node, calculate the tentative distance to that neighbor by adding the weight of the edge connecting the current node to the neighbor to the tentative distance of the current node. If the tentative distance is less than the current distance assigned to the neighbor, update the neighbor's distance to the tentative distance.
3. Add the neighbor to the priority queue if you haven't already
4. Repeat until the destination node has been removed from the priority queue or the priority queue is empty, at which point the algorithm will have completed its job .

And if you want to find the shortest path back to the source node all the way from the destination node, all you have to do is backtrack from the destination node using the "parent" information that was stored during the algorithm. The priority queue used in Dijkstra's algorithm is usually implemented using a heap data structure, which allows for efficient insertion and removal of elements based on their priority, which is determined by their distance to the start node.

Important note is Dijkstra's algorithm only works correctly on graphs with non-negative edge weights.

Prim's Algorithm, designed by Vojtěch Jarník, is a greedy algorithm used to find the minimum spanning tree of a weighted undirected graph. The minimum spanning tree is a subset of the edges of the graph that connects all the vertices together without any cycles and has the minimum possible total edge weight. This algorithm is put together by first picking a random vertex and adding it to the MST. Next, find the smallest weight edge that connects the MST to a vertex not yet inside it. Then, add the vertex that weight is connected to to the MST. Then repeat until all vertices have been added. This algorithm is commonly implemented using a priority queue data structure for efficiency, and the time efficiency of this algorithm is O(ElogV) where E are the number of edges and V is the number of vertices.