Dijkstra's Algorithm:

- Working:

Dijkstra's Algorithm finds the shortest path from a starting node to all other nodes in a weighted graph. It maintains a list of visited nodes and updates the distance from the start node to each unvisited neighbor. It continually selects the node with the smallest tentative distance and explores its neighbors.

- Key Points:

Dijkstra's Algorithm guarantees the shortest path but can be computationally expensive, especially for large graphs. It's suitable for scenarios where optimality is critical, like network routing.

Wavefront Algorithm:

- Working:

The Wavefront Algorithm is a grid-based path planning method. It assigns cost values to cells starting from the goal and propagates costs outward. It computes the optimal path by moving from the starting cell to adjacent cells with lower cost values.

- Key Points:

It's efficient for grid-based maps and simple to implement. However, it doesn't work well in continuous environments or when obstacles have irregular shapes.

A* Algorithm:

- Working:

A* is an informed search algorithm that combines Dijkstra's approach with a heuristic function. It explores nodes by considering both the actual cost from the start node and an estimate of the remaining cost to the goal. It selects nodes with the lowest combined cost.

- Key Points:

A* is widely used for path planning due to its efficiency and optimality. It's suitable for grid-based and continuous environments and offers a good balance between completeness and speed.

D* Lite:

- Working:

D* Lite is an incremental search algorithm designed for dynamic environments. It starts with a known path and incrementally updates it as the environment changes. It focuses on re-exploring only parts of the path affected by changes.

- Key Points:

D* Lite is efficient for scenarios where the environment may change over time. It can quickly adapt to new obstacles or information updates while minimizing redundant path re-computation.