**Topic:**

**‘A\* Algorithm’**

The A\* algorithm is a popular pathfinding algorithm used to find the shortest path between two nodes in a weighted graph. It is an extension of Dijkstra's algorithm that uses a heuristic function to guide the search towards the goal node, making it more efficient.

## Functions:

The A\* algorithm implementation consists of the following key functions:

### 1. \*\_\_init\_\_\*

- Initializes the graph object with a list of nodes and their corresponding heuristic values.

- The list is a dictionary where each key is a node and its value is a list of tuples containing the neighboring nodes and their weights.

- The heuristic map is a dictionary where each key is a node and its value is the estimated cost to reach the goal node.

### 2. \*neighbors\*

- Returns a list of neighboring nodes for a given node.

- The neighboring nodes are retrieved from the graph's list dictionary.

### 3. \*h\*

- Returns the heuristic value for a given node.

- The heuristic value is retrieved from the graph's heuristic map.

### 4. \*star\_alg\*

- Implements the A\* algorithm to find the shortest path between a start node and a goal node.

- The algorithm uses a priority queue (implemented using a set) to keep track of nodes to be evaluated.

- It maintains two dictionaries, g and p, to store the cost of reaching each node and its parent node, respectively.

- The algorithm iteratively selects the node with the lowest f-score (cost + heuristic) and explores its neighbors until the goal node is reached or the open list is empty.

## How it Works:

1. Initialize the graph object with a list of nodes and their corresponding heuristic values.

2. Call the star\_alg function with the start node and goal node as arguments.

3. The algorithm will return the shortest path from the start node to the goal node if one exists.

## Key Concepts:

- Heuristic Function: An estimate of the cost to reach the goal node from a given node.

- Cost Function: The actual cost of reaching a node from the start node.

- F-Score: The sum of the cost function and the heuristic function.

- Open List: A priority queue of nodes to be evaluated.

- Closed List: A set of nodes that have been evaluated.

## Advantages:

- Efficient: A\* algorithm is more efficient than Dijkstra's algorithm when a good heuristic function is used.

- Optimal: A\* algorithm guarantees to find the shortest path to the goal node if the heuristic function is admissible.

## Applications:

- Pathfinding: A\* algorithm is widely used in video games, robotics, and other applications where pathfinding is required.

- Route Planning: A\* algorithm can be used to find the shortest route between two points on a map.