

1. Read about the general operation of the following search strategies for both tree search and graph search
 1. Depth first search
 2. Breadth First Search
 3. Uniform cost search
 4. Greedy search
 5. A* search

Answers

1. Depth First Search

Operation:

It explores as far as possible along a branch before backtracking.

Tree Search:

In a tree search, DFS uses stacks data structure to keep track of nodes to visit.

Graph Search:

In graph search, DFS maintains a set of visited nodes to avoid revisiting and pushing unvisited neighbors onto the stack.

2. Breadth First Search

Operation:

It explores neighbors at a current depth level before moving to the next level.

Tree Search:

In a tree search, BFS uses queues data structure to visit node in a level-wise manner.

Graph Search:

In graph search, BFS maintains a set of visited nodes to avoid revisiting and enqueue unvisited neighbors.

3. Uniform Cost Search

Operation:

It selects the path with the lowest cost.

Tree Search:

In a tree search, UCS uses priority queue or min-heap based on cumulative path cost.

Graph Search:

In graph search, UCS maintains a set of visited nodes and updates the cost of a lower-cost path till a visited node is found.

4. Greedy Search

Operation:

It selects the path that appears to be the best based on a heuristic while ignoring the path cost.

Tree Search:

In a tree search, Greedy search uses priority queue based on the heuristic estimate.

Graph Search:

In graph search, Greedy search operates like in tree search but maintains a set of visited nodes to avoid loops.

5. A* Search

Operation:

A* combines the cost so far and heuristic to select the most promising path.

Tree Search:

In a tree search, A* search uses a priority queue based on the sum of the cost and heuristic estimate.

Graph Search:

In graph search, A* search operates similar to tree search but maintains a set of visited nodes and updates the cost if a lower-cost path is found