## Name:

## Informed Search Algorithms

### 1. Define and explain the following concepts:

#### a. What is an Informed Search Algorithm?

An **informed search algorithm** is a type of algorithm that uses extra information, called a heuristic, to help it find the best path or solution more quickly. Unlike uninformed search algorithms, which explore all possibilities without any guidance, informed search algorithms use heuristics to guess which paths are likely to lead to the goal. This helps the algorithm focus on the best paths and avoid wasting time on less promising ones.

#### b. How Does A\* Search Differ from Dijkstra's Algorithm and Greedy Best-First Search?

The **A\*** search algorithm is a mix of two other algorithms: **Dijkstra’s algorithm** and **Greedy Best-First Search**:

* **Dijkstra’s Algorithm**:
  + It looks for the shortest path from the starting point to all other points, considering the total cost to get to each point.
  + It doesn’t use any heuristics, just the cost of the path so far.
* **Greedy Best-First Search**:
  + It uses a heuristic to guess how close a point is to the goal but doesn’t consider the cost to get to that point.
  + This can lead to bad paths because it focuses only on getting close to the goal quickly, not on the overall best path.
* **A\***:
  + A\* combines both approaches by considering both the cost of the path so far and an estimate of the remaining cost to the goal.
  + It uses a formula to evaluate each point: , where is the cost to get from the start point to node , and is the estimated cost from node to the goal.
  + This helps A\* find the best possible path more efficiently.

#### c. Define and Explain Components g(n), h(n), and f(n) in A\* Search

In A\* search, the three main parts are:

* : This is the actual cost to reach the current node from the start. It keeps track of how much has been spent so far.
* : This is the heuristic, which estimates how much more it will cost to reach the goal from node .
* : This is the total estimated cost of the best path through node to the goal. It’s calculated like this:   
    
  This helps A\* decide which node to explore next by looking at both the path cost so far and the estimated cost to the goal.

### 2. Answer the following questions:

#### a. What Conditions Must Be Met for A\* Algorithm to Guarantee an Optimal Solution?

For A\* to always find the best possible solution, two conditions need to be met:

1. The heuristic must be **admissible**, meaning it should never overestimate the true cost to get from node to the goal.
2. The heuristic must also be **consistent** (or monotonic), which means that for any node and its successor , the heuristic should satisfy:  
     
   where is the actual cost to move from node to .

#### b. Explain Why the Heuristic Function Must Be "Admissible" in A\* Search

The heuristic must be admissible because:

* If it overestimates the cost to the goal, A\* might skip the best path, leading to a wrong or suboptimal solution.
* An admissible heuristic ensures that A\* will always find the best solution if one exists and will do so efficiently.

#### c. Describe Three Real-World Applications Where A\* Search Would Be Particularly Useful

A\* search is used in many different real-world situations:

1. **Pathfinding in Video Games**: A\* is used to guide characters or non-playable characters (NPCs) in video games, helping them find the best route across the game world.
2. **Robotics Navigation**: Robots use A\* to plan their paths and avoid obstacles while moving through an environment.
3. **GPS Navigation Systems**: A\* can help GPS systems find the best driving route by taking into account factors like road distance, traffic, and other conditions.