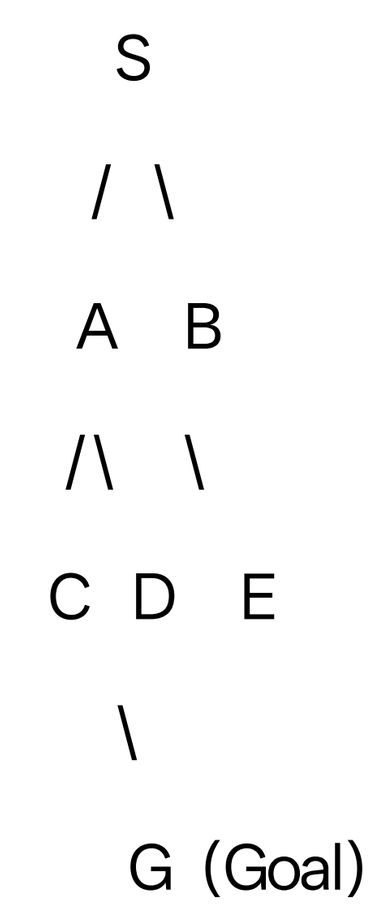
AIML lab assignment 1

Problem Description (Common for All Tasks)

Consider the following state-space graph representing a search problem.

Each node represents a state and edges represent possible actions.

Graph Structure



Heuristic Values h(n)

Node	h(n)
S	6
A	4
B	5

C 7

D 2

E 6

G 0

Edge Costs (for A*)

Edge Cost

S → A 1

S → B 2

A → C 3

A → D 1

B → E 4

D → G 2

PRACTICAL TASKS

Task 1: Breadth First Search (BFS)

Question:

Write a Python program to implement Breadth First Search to find the

shortest path from the start node S to the goal node G.

Expected Output:

Traversal order

Path from S to G

Task 2: Depth First Search (DFS)

Question:

Implement Depth First Search using recursion or stack to explore the graph starting from node S and stopping when node G is reached.

Expected Output:

DFS traversal

Path obtained

Task 3: Best First Search

Question:

Develop a Python program to implement Best First Search using the given heuristic values to select the next node for expansion.

Expected Output:

Order of node expansion

Path from S to G

Task 4: A* Search Algorithm

Question:

Implement the A* search algorithm using the evaluation function:

$$f(n) = g(n) + h(n)$$

where:

is the path cost from start node

is the heuristic estimate

Expected Output:

Optimal path from S to G

Total path cost

Task 5: Hill Climbing Algorithm

Question:

Write a Python program to implement the Hill Climbing algorithm for the given problem.

Use heuristic values to move towards the goal state.

Expected Output:

Sequence of states visited

Final state reached

Observation on local maxima or plateaus (if any)

Task 6: Simulated Annealing Algorithm

Question:

Implement the Simulated Annealing algorithm to solve the same problem and overcome the limitations of Hill Climbing.

Expected Output:

Sequence of states visited

Final solution

Effect of temperature and cooling schedule

Some theory questions

1. Why does BFS guarantee the shortest path?

2. Why DFS is memory efficient but not optimal?

3. How does heuristic influence Best First Search?

4. Why is A* considered complete and optimal?

5. What are the drawbacks of Hill Climbing?

6. How does Simulated Annealing escape local maxima?