Study Guide: Introduction to Artificial Intelligence

Search

* **Motivation and Outline**

□+ Problem Formulation and Examples (Romania problem, deterministic, fully observable)

* **Basic Tree Search Algorithms**

□+ Tree Search Implementation: States vs Nodes

□+ Tree Search: General Algorithm

□+ **Breadth-First Search (BFS))**

□□- Definition: "Cost-aware BFS": pick least-cost unexpanded node

□□- Implementation: fringe = queue ordered by path cost, lowest first

□□- Equivalent to breadth-first if step costs all equal

□□- **Complete?**: Yes, if step cost ≥ ε

□□- **Time?**: # of nodes with g ≤ cost-of-optimal-solution, O(b^ ⌈C*/ε⌉)

□□- **Space?**: # of nodes with g ≤ cost-of-optimal-solution, O(b^ ⌈C*/ε⌉)

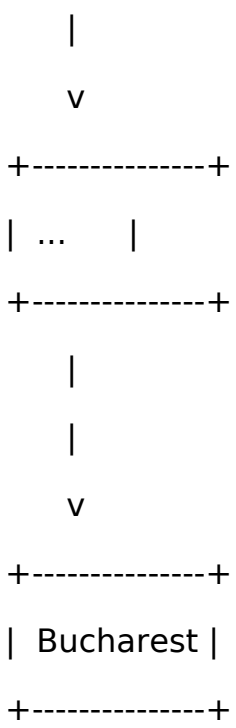
□□- **Optimal?**: Yes: nodes expanded in increasing order of g(n)

Example:

Romania problem: search from Lugoj to Bucharest using BFS.

Diagram:

```
    +---------------+
    |  Lugoj    |
    +---------------+
        |
```



Summary of Key Points

• Breadth-first search (BFS) is a tree search algorithm that expands nodes in increasing order of cost (or distance) from the starting node.

• BFS is complete and optimal if step costs are all equal or if the step cost is at least ε.

• BFS has a time complexity of O(b^ ⌈C*/ε⌉), where C* is the cost of the optimal solution, and a space complexity of O(b^ ⌈C*/ε⌉).

Flashcards:

Q1: What is the Breadth-First Search (BFS) algorithm?

A1: BFS is a tree search algorithm that expands nodes in increasing order of cost (or distance) from the starting node.

Q2: Is BFS complete?

A2: Yes, BFS is complete if step costs are all equal or if the step cost is at least ε.

Q3: What is the time complexity of BFS?

A3: The time complexity of BFS is O(b^ ⌈C*/ε⌉), where C* is the cost of the optimal solution.

Q4: What is the space complexity of BFS?

A4: The space complexity of BFS is O(b^ ⌈C*/ε⌉).

Q5: Is BFS optimal?

A5: Yes, BFS is optimal if step costs are all equal or if the step cost is at least ε.