**Week 4 Lecture Notes**

Search Problems:

State Space

Successor function:

Start State and Goal State:

Solution: Sequence of actions.

**8-Puzzle State Space Search Example:**

Diagram

Description automatically generated with medium confidence

**States**: Location of tiles (Initial State could be any state)

**Actions**: move blank, left, right, up, down

**Transitional Model**: Given a State and Action it returns resulting Space

**Goal Test**: Goal State (Given)

**Path Cost**: 1 per move

**Search algorithms can be classified as.**

**Informed Search**: Greedy Search, A\* Search, Graph Search

**Uninformed Search**: Depth First Search, Breadth First Search, Uniform Cost Search

|  |  |  |
| --- | --- | --- |
| Basic Comparison | Informed Search | Uninformed |
| Basic Knowledge | Uses Knowledge to find the steps to the solution | No use of Knowledge |
| Efficiency | Highly efficient as it consumes less time and cost. | Efficiency is mediatory |
| Cost | Low | Comparatively High |
| Performance | Finds solution quick | Slower speed than the informed search |

**Depth first search**

If DFS (m) is finite the time taken O(b^m)

You expand to the deepest unexpanded node until you fail or find a solution.

Deepest = LIFO (Last in first out) queue

A picture containing accessory

Description automatically generatedStarts with A goes to B and keeps expanding from the left and keeps expanding down, once it finished expanding until it cant anymore (H) then it goes to the next one (I) and if it cant then it goes back up (to E) and Continues going.

A picture containing accessory, toggle, necklet

Description automatically generated

**Breadth First Search,**

Expand the shallowest node until you fail or find a solution.

Shallowest = FIFO Queue

Shallower nodes are always selected first

A picture containing diagram

Description automatically generatedNew nodes go to back of the queue

This is a level-by-level search, Search in tiers. Search take time O(b^d).

Table

Description automatically generatedD must be finite if a solution exists.