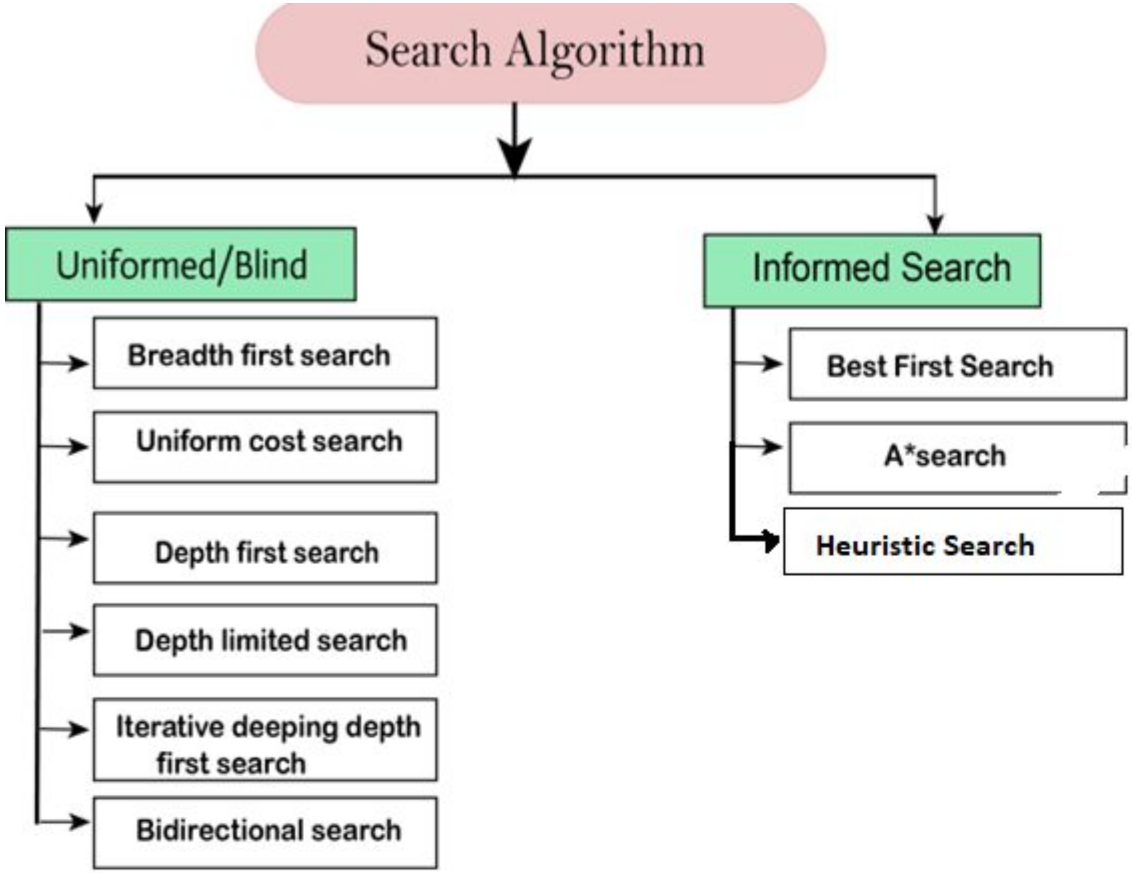
State Space Representation & Heuristic Search

State Space Representation of Problems

- Before an AI problem can be solved it must be represented as a state space.
- The state space is then searched to find a solution to the problem.
- Node:
  - A state space essentially consists of a set of nodes representing each state of the problem
- Arc:
  - Arcs between nodes representing the legal moves from one state to another, an initial state and a goal state.
- Each state space takes the form of a tree or a graph.

- Factors that determine which search algorithm or technique will be used include the
  - Type of the problem and
  - How the problem can be represented.



Classic AI Problems

- Travelling salesman problem
- Towers of Hanoi
- 8-Puzzle

Problem Representation

- A number of factors need to be taken into consideration when developing a state space representation:
  - What is the goal to be achieved?
  - What are the legal moves and actions?
  - What knowledge needs to be represented in the state description?
  - Type of Problem
    - That need only representation- crossword puzzles
    - That require a yes or no response indicating whether a solution can be found or not
    - That require a solution path as an output, e.g. mathematical theorems, Tower of Hanoi.
      - In this case we need to know the goal state and we need to know how to attain this state.
  - Best solution vs. Good enough solution
    - For some problems a good enough solution is sufficient for e.g. Theorem proving, eight squares
    - However, some problems require a best solution, e.g. the travelling salesman problem.

State Space Representation- Tower of Hanoi

- A possible state space representation using graph
- The legal moves in this state space involve moving one ring from one pole to another, moving one ring at a time, and ensuring that a larger ring is not placed on a smaller ring.

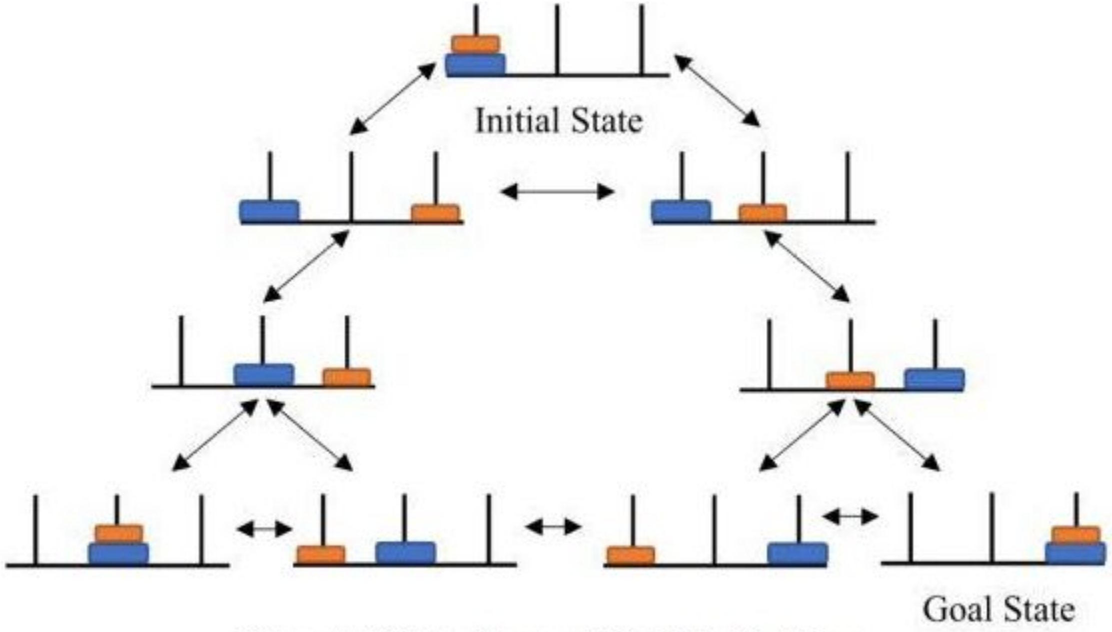
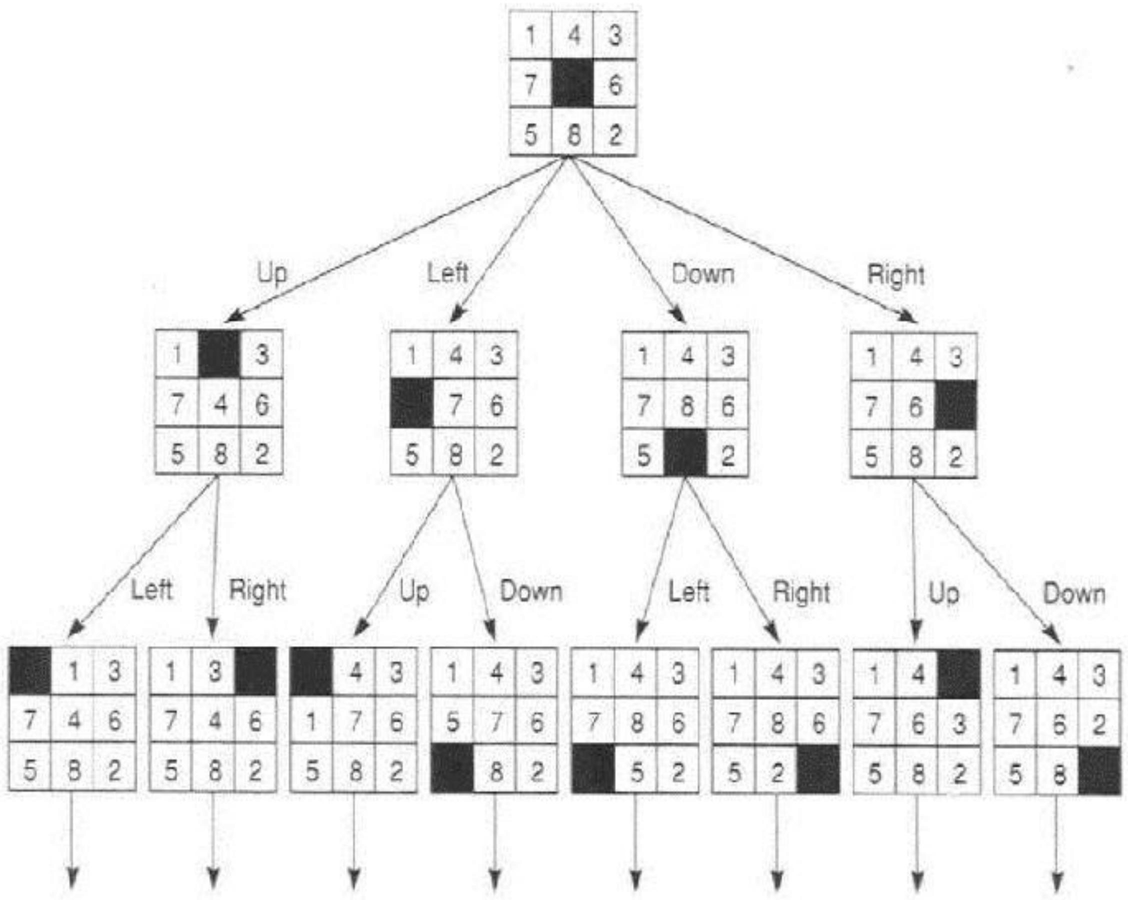


Figure 2 State Space of 2-Disk Problem

State Space Representation: 8-puzzle

- Although a player moves the tiles around the board to change the configuration of tiles.
- However, we define the legal moves in terms of moving the space.
  - The space can be moved up, down, left and right.



Summary

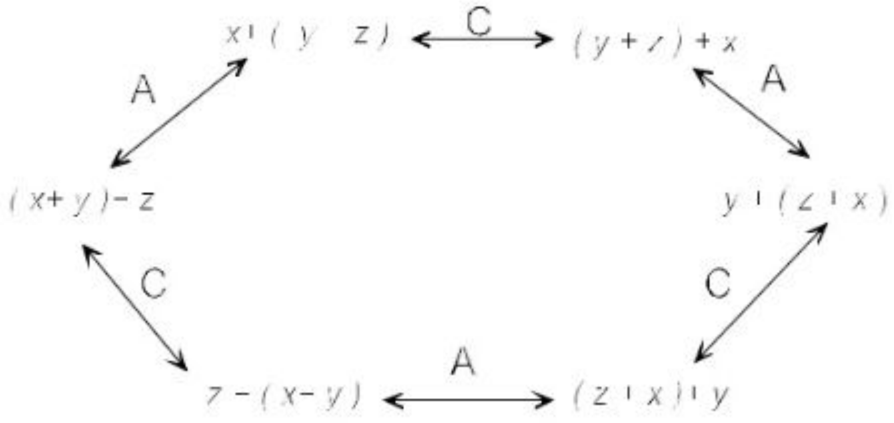
- A state space is a set of descriptions or states.
- Each search problem consists of
    - One or more initial states
    - A set of legal actions
        - Actions are represented by operators or moves applied to each state. For example operators in state space representation of the 8-puzzle problem are left, right, up and down.
    - One or more goal states
- The number of operators are problem dependent and specific to a particular state space representation.
    - The more operators the larger the branching factor of the state space
    - Thus, the number of operators should kept to a minimum, e.g. in 8-puzzle problem, operations are defined in terms of moving the space instead of the titles.
- A search algorithm is applied to a state space representation to find a solution path.
    - Each search algorithm applies a particular search strategy.

Graph versus Trees

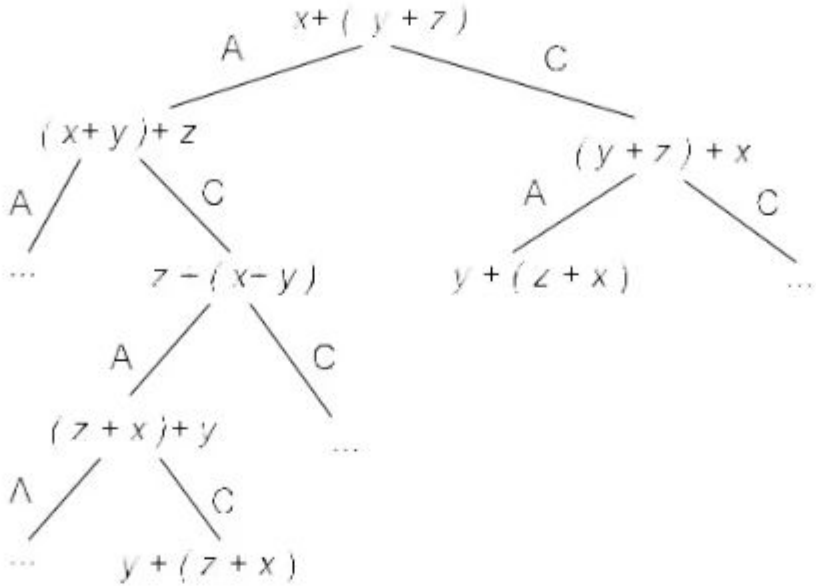
- A directed graph can be used to represent the solution space
  - If states in the solution space can be revisited more than once
  - In a graph more than one move sequence can be used to get from one state to another
  - In a graph there is more than one path to a goal whereas in a tree a path to a goal is more clearly distinguishable.
- In tree
  - A goal state may need to appear more than once.
- Search algorithms for graphs have to cater for possible loops and cycles in the graph so trees may be more efficient for representing such problems as loops and cycles do not have to be catered for.
- The entire tree or graph will not be generated.

Example

- Prove x + (y + z) = y + (z + x) given
    - L + (M + N) = (L + M) + N
    - M + N = N + M



Graph state space representation



Tree state space representation

Search Algorithm Terminologies

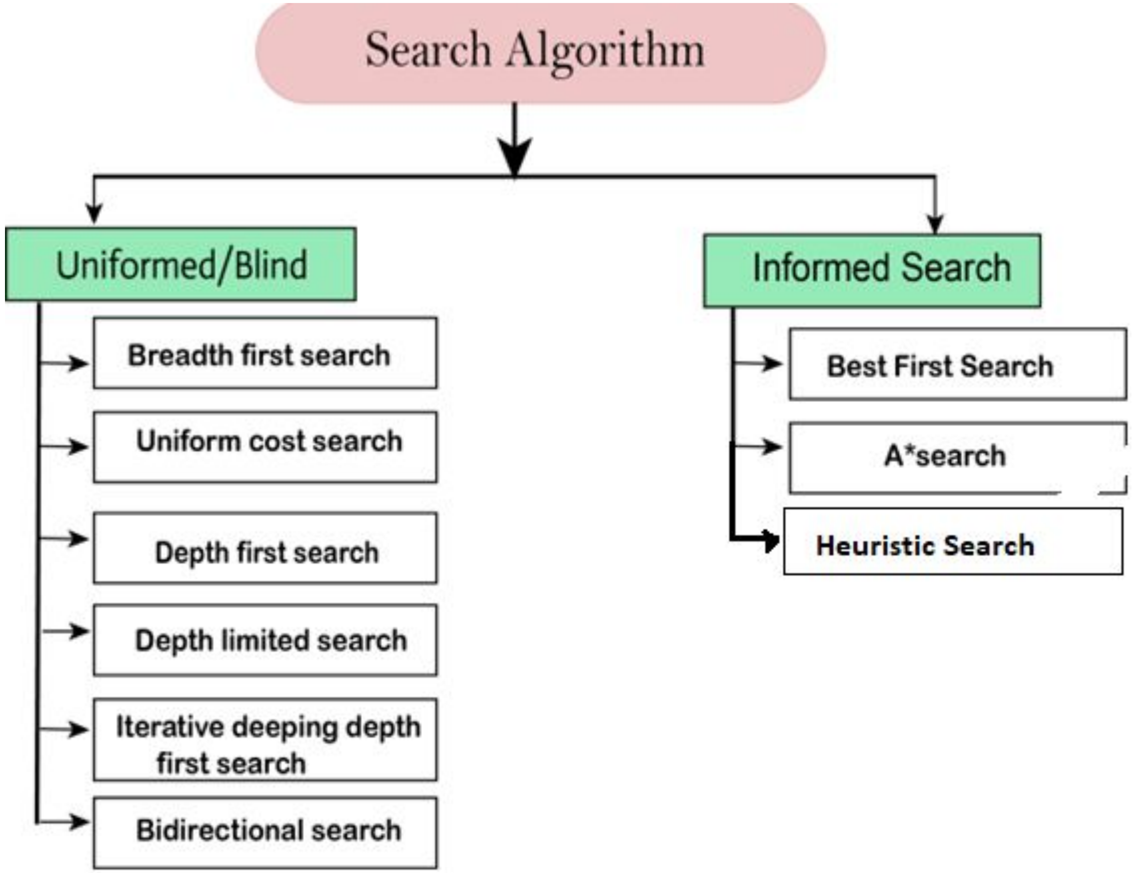
- **Search:** Searchings a step by step procedure to solve a search-problem in a given search space. A search problem can have three main factors:
  - **Search Space:** Search space represents a set of possible solutions, which a system may have.
  - **Start State:** It is a state from where agent begins **the search**.
  - **Goal test:** It is a function which observe the current state and returns whether the goal state is achieved or not.
- **Search tree:** A tree representation of search problem is called Search tree. The root of the search tree is the root node which is corresponding to the initial state.
- **Actions:** It gives the description of all the available actions to the agent.
- **Transition model:** A description of what each action do, can be represented as a transition model.
- **Path Cost:** It is a function which assigns a numeric cost to each path.
- **Solution:** It is an action sequence which leads from the start node to the goal node.
- **Optimal Solution:** If a solution has the lowest cost among all solutions.

Properties of Search Algorithms

- Following are the four essential properties of search algorithms to compare the efficiency of these algorithms:
- **Completeness:** A search algorithm is said to be complete if it guarantees to return a solution if at least any solution exists for any random input.
- **Optimality:** If a solution found for an algorithm is guaranteed to be the best solution (lowest path cost) among all other solutions, then such a solution for is said to be an optimal solution.
- **Time Complexity:** Time complexity is a measure of time for an algorithm to complete its task.
- **Space Complexity:** It is the maximum storage space required at any point during the search, as the complexity of the problem.

Types of search algorithms

- Based on the search problems we can classify the search algorithms into
  - uninformed (Blind search) search and
  - informed search (Heuristic search) algorithms.



Uninformed/Blind Search

- **is a class of general-purpose search algorithms which operates in brute force-way**
- The uninformed search does not contain any domain knowledge such as closeness, the location of the goal.
- **It knows only how to traverse the tree**
- It examines each node of the tree until it achieves the goal node.

Informed Search Algorithms

- Informed search algorithms use domain knowledge.
- In an informed search, problem information is available which can guide the search.
- Informed search strategies can find a solution more efficiently than an uninformed search strategy.
- Informed search is also called a Heuristic search.
- A heuristic is a way which might not always be guaranteed for best solutions but guaranteed to find a good solution in reasonable time.