DRONACHARYA College of Engineering

INTELLIGENT SYSTEMS (CSE-303-F)

Section A

Problems and Search

Outline

- State space search
- Search strategies
- Problem characteristics
- Design of search programs

State Space Search

Problem solving = Searching for a goal state

State Space Search: Water Jug Problem

"You are given two jugs, a 4-litre one and a 3-litre one. Neither has any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 litres of water into 4-litre jug."

State Space Search: Water Jug Problem

- State: (x, y)

  x = 0, 1, 2, 3, or 4 y = 0, 1, 2, 3

- Start state: (0, 0).

- Goal state: (2, n) for any n.

- Attempting to end up in a goal state.

State Space Search: Water Jug Problem

1. (x, y) if $x < 4$ $\rightarrow (4, y)$ [fill the 4 ltr jug]
2. (x, y) if $y < 3$ $\rightarrow (x, 3)$ [fill the 3 ltr jug]
3. (x, y) if $x > 0$ $\rightarrow (x - d, y)$ [pour some water out of 4 ltr jug]
4. (x, y) if $y > 0$ $\rightarrow (x, y - d)$ [pour some water out of 3 ltr. Jug]

State Space Search: Water Jug Problem

5. (x, y) $\rightarrow (0, y)$ [empty 4 ltr jug]
   if $x > 0$
6. (x, y) $\rightarrow (x, 0)$ [empty 3 ltr. jug]
   if $y > 0$
7. (x, y) $\rightarrow (4, y - (4 - x))$

   if $x + y \geq 4, y > 0$

   [pour water from 3 ltr jug in 4 ltr jug until the 4 ltr. Jug is full]
8. (x, y) $\rightarrow (x - (3 - y), 3)$
   if $x + y \geq 3, x > 0$
   [pour water from 4 ltr jug in 3 ltr jug until the 3 ltr. Jug is full]

State Space Search: Water Jug Problem

9. $(x, y) \rightarrow (x + y, 0)$
   if $x + y \le 4,\ y > 0$
   [pour all water from 3 ltr jug in 4 ltr. Jug]

10. $(x, y) \rightarrow (0, x + y)$
    if $x + y \le 3,\ x > 0$

    [pour water from 4 ltr. Jug in 3 ltr. Jug]

11. $(0, 2) \rightarrow (2, 0)$

    [pour 2 ltr from 3 ltr jug to 4 ltr jug]

12. $(2, y) \rightarrow (0, y)$
    [pour 2 ltr from 3 ltr jug to 4 ltr jug]

State Space Search: Water Jug Problem

1. current state = (0, 0)
2. Loop until reaching the goal state (2, 0)
   - Apply a rule whose left side matches the current state
   - Set the new current state to be the resulting state

(0, 0)	
(0, 3)	rule applied 2
(3, 0)	rule applied 9
(3, 3)	rule applied 2
(4, 2)	rule applied 7
(0, 2)	rule applied 5
(2, 0)	rule applied 9

State Space Search: Summary

1. Define a state space that contains all the possible configurations of the relevant objects.
2. Specify the initial states.
3. Specify the goal states.
4. Specify a set of rules:
    - What are unstated assumptions?
    - How general should the rules be?
    - How much knowledge for solutions should be in the rules?

Search Strategies

Requirements of a good search strategy (to decide which rule to apply next during process of searching for a solution to the problem):

1. It causes motion

   Otherwise, it will never lead to a solution.

2. It is systematic

   Otherwise, it may use more steps than necessary.

3. It is efficient

   Find a good, but not necessarily the best, answer.

Search Strategies

1. Uninformed search (blind search/brute force search)

   Having no information about the number of steps from the current state to the goal.

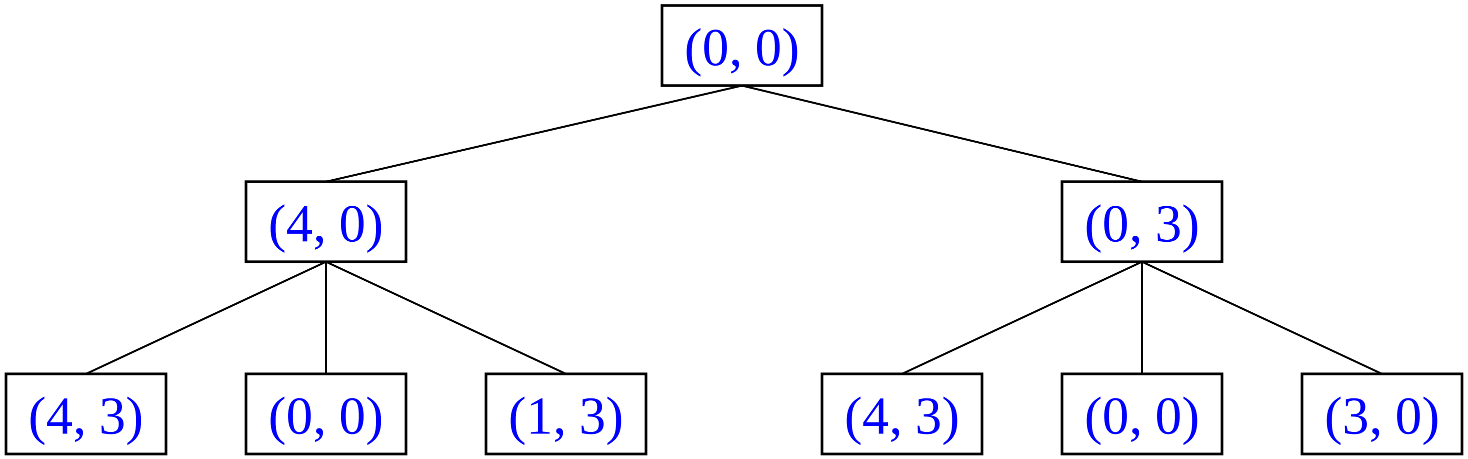
   - Depth-First Search
   - Breadth-First Search

2. Informed search (heuristic search)

   More efficient than uninformed search.

   - Best-First Search
   - Hill Climbing
   - A*

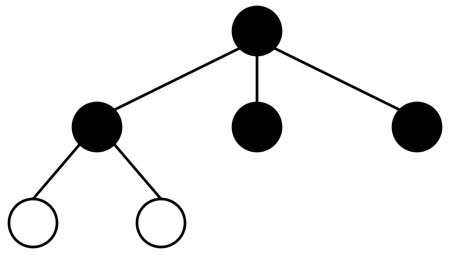
Search Strategies



Search Strategies: Blind Search

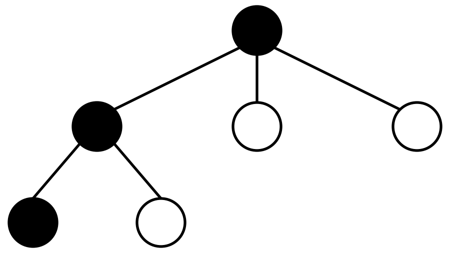
- Breadth-first search

  Expand all the nodes of one level first.



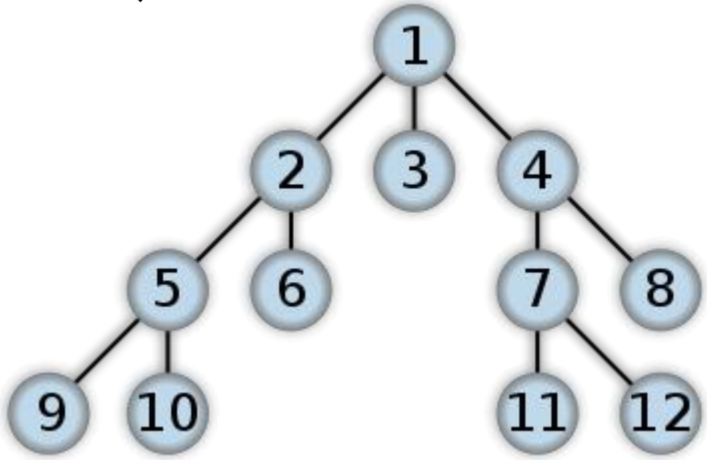
- Depth-first search

  Expand one of the nodes at the deepest level.



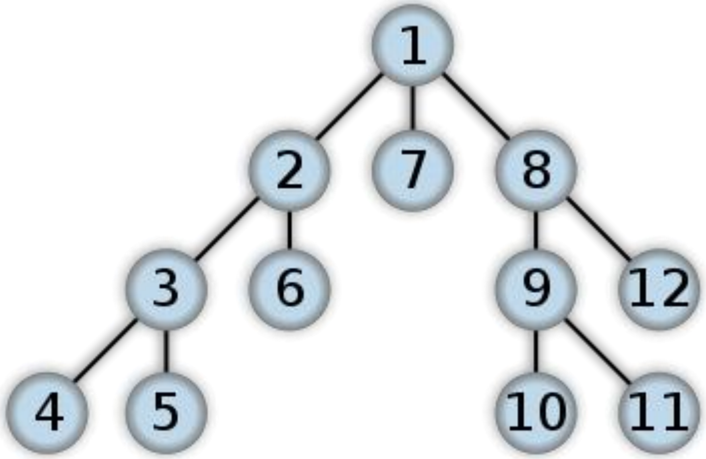
Breadth First Search

- Start at the root node
- Scan each node in the first level starting from the leftmost node, moving towards the right.
- Continue scanning the second level (starting from the left) and the third level, and so on.



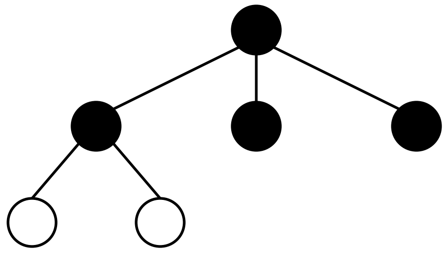
Depth First Search

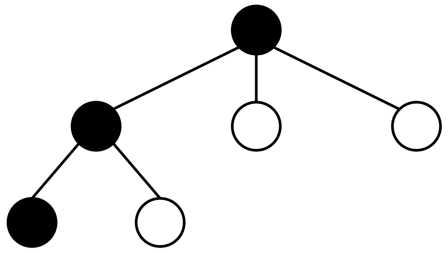
- Start at the root
- Follow one of the branches of the tree as far as possible until either the node you are looking for is found or you hit a leaf.
- If you hit a leaf node, then you continue the search at the nearest ancestor with unexplored children.



Search Strategies: Blind Search

Criterion	Breadth-First	Depth-First
Time		
Space		
Optimal?		
Complete?		

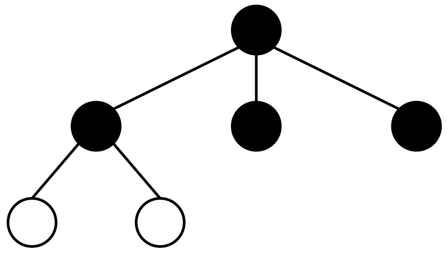


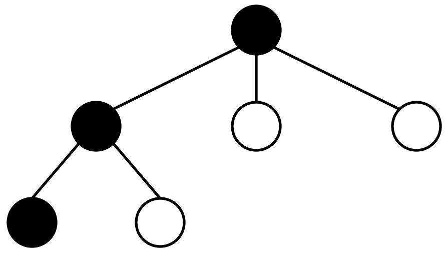


b: branching factor d: solution depth m: maximum depth

Search Strategies: Blind Search

Criterion	Breadth-First	Depth-First
Time	b^d	b^m
Space	b^d	bm
Optimal?	Yes	No
Complete?	Yes	No





b: branching factor d: solution depth m: maximum depth

Search Strategies: Heuristic Search

- Heuristic: involving or serving as an aid to learning, discovery, or problem-solving by experimental and especially trial-and-error methods.
  (Merriam-Webster's dictionary)
- Heuristic technique improves the efficiency of a search process, possibly by sacrificing claims of completeness or optimality.

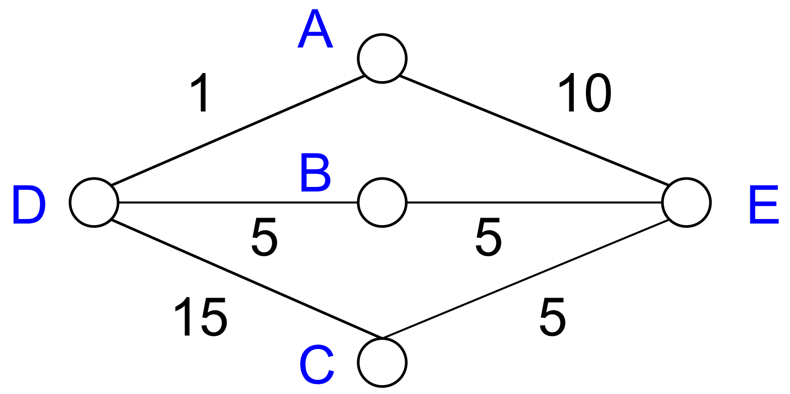
Search Strategies: Heuristic Search

- Heuristic is for combinatorial explosion.
- Optimal solutions are rarely needed.

Search Strategies: Heuristic Search

The Travelling Salesman Problem

"A salesman has a list of cities, each of which he must visit exactly once. There are direct roads between each pair of cities on the list. Find the route the salesman should follow for the shortest possible round trip that both starts and finishes at any one of the cities."



Search Strategies: Heuristic Search

Nearest neighbour heuristic:

1. Select a starting city.
2. Select the one closest to the current city.
3. Repeat step 2 until all cities have been visited.

Search Strategies: Heuristic Search

- Heuristic function:

  state descriptions $\rightarrow$ measures of desirability

Problem Characteristics

To choose an appropriate method for a particular problem:

- Is the problem decomposable?
- Can solution steps be ignored or undone?
- Is the universe predictable?
- Is a good solution absolute or relative?
- Is the solution a state or a path?
- What is the role of knowledge?
- Does the task require human-interaction?

Is the problem decomposable?

- Can the problem be broken down to smaller problems to be solved independently?
- Decomposable problem can be solved easily.

Is the problem decomposable?

$$\begin{tikzcd}
 & \int(x^2 + 3x + \sin^2x.\cos^2x)dx \arrow[dl, dash] \arrow[d, dash] \arrow[dr, dash] & & \\
\int x^2dx & \int 3xdx & \int\sin^2x.\cos^2xdx \arrow[d, dash] & \\
 & & \int(1 - \cos^2x)\cos^2xdx \arrow[dl, dash] \arrow[dr, dash] & \\
 & \int\cos^2xdx & & -\int\cos^4xdx
\end{tikzcd}$$

Can solution steps be ignored or undone?

Theorem Proving

A lemma that has been proved can be ignored for next steps.

Ignorable!

Can solution steps be ignored or undone?

The 8-Puzzle

2	8	3
1	6	4
7		5

$\Rightarrow$

1	2	3
8		4
7	6	5

Moves can be undone and backtracked.

Recoverable!

Can solution steps be ignored or undone?

- Ignorable problems : solution steps can be ignored. Eg. Theorem proving
- Recoverable problems : solution steps can be undone. Eg. 8 puzzle

- Irrecoverable problems : solution steps can’t be undone. Eg chess

Is the universe predictable?

The 8-Puzzle

Every time we make a move, we know exactly
what will
happen.

Certain outcome!

What is the role of knowledge

Playing Chess

Knowledge is important only to constrain the search for
a solution.

Reading Newspaper

Knowledge is required even to be able to recognize a
solution.

Problem Classification

- There is a variety of problem-solving methods, but there is no one single way of solving all problems.

- Not all new problems should be considered as totally new. Solutions of similar problems can be exploited.