Unit:-2

Problem Solving and Representation of Knowledge

19/2

* Problem Solving Agents: Goal-Based
=> They are goal-based agents.
=> One of the goal -> to maximize performance measures.
=> Goal formation
=> Problem formulation
=> Search

* Problem formulation is the process of deciding what actions and state to consider given a goal.

* The process of looking for sequence of actions that leads an agent to the goal is known as search.
=> search Algorithm takes a problem as input and returns soln having sequence of actions.

=> Problem formulation :-
- Initial state - the starting state
- Action
- Transition model - description of action results
- Goal test - Is Goal?
- Path cost - The fun^n returns heuristic value used to estimate the performance measures.

* Problem Formulation of Vacuum Cleaner World.

1> States : $2 \times 2^2 = 8$ (total)

2) Initial state : Any state from the available states
3) Actions : Shift left, shift right, suck the dirt
4) Transition model :
5) Goal Test :
6) Path Cost : 1

* 8 - Puzzle Problem (Problem formulation)

7	2	4
5		6
8	3	1

Start state

	1	2
3	4	5
6	7	8

Goal state

* 8 - Puzzle Problem

States :- 81

Initial :- All possible state with no two queen in same row, column or diagonal

Actions :- Moving queen to different col in its row

Transitional model :- stage change when queen is moved to new position

Goal Test :- check no two queen can attack each other.

Path cost :- No. of steps to reach solution.

Root finding Problem

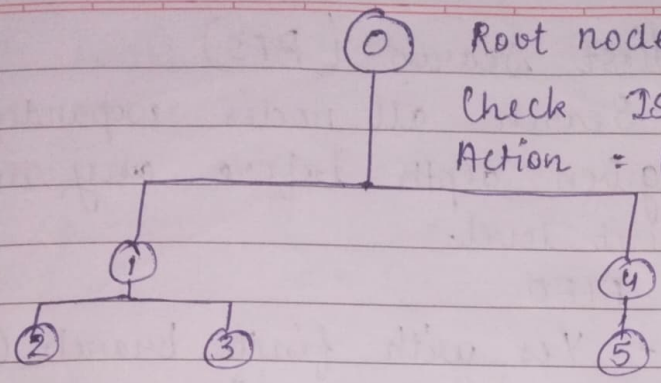
1) States :- Diff. values of x
2) Initial state :- selection of x
3) Actions :- Adjusting the guess for x based on finding method.
4) Transitional model :- How the value of x is updated.
5) Goal Test :- Root is found
6) Path cost :- No. of iterations or stips to reach the root.

Robot Navigation

1) States :-
2) Initial state :- Starting position of robot in environment.
3) Actions :- The movements that robot can make.
4) Transition model :- How robot's state change upon actions.
5) Goal Test :- Checking if robot has reached its destination.
6) Path cost :- Cumulative cost to reach goal.

☆ Searching for Solution

- Solution ? (Sequence of actions)
- Select Initial state
- Apply action & generate new state
- Check for goal & repeat
- return solution



Measures :-

- Completeness (Is the sol^n guaranteed ?)
- Optimality (Optimal sol^n ?)
- Time Complexity
- Space Complexity

Searching for Sol^n (Search strategies)

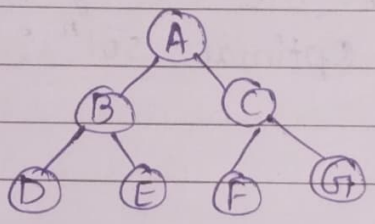
Blind Search	Heuristic Search
- Breadth First Search	- A * Algo
- Depth First Search	- Greedy BFS.

* Blind Search :- It is also known as uninformed search ~~categories~~ strategies.
- It traverse the search space until goal node is found.

* Heuristic :- Also known as informed search strategies.
- Search Process takes place by traversing search space with applied rules / information.

☆ Breadth-First Search (BFS)

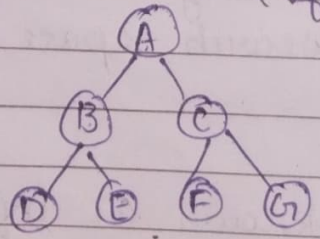
- Strategy - Search all nodes expanded at given depth before any node at next level.
- Concept - FIFO
- Complete ? - Yes with finite branch (b)
- Space - Keep every node in memory
- optimal ? - Yes (If cost = 1)



Path :- ~~D-E-B-F-G-C-A~~ A-B-C-D-E-F-G

☆ DFS

- Strategy :- Search all nodes expanded at given depth before any node at deepest path.
- Concept :- FILO
- Complete ? :- ~~Yes~~ No with infinite branch (b)
- space - :- Keep every node in memory
- optimal ? :- ~~Yes~~ No (If cost = 1)



Path :- A-B-D-E-C-F-G

★ Depth Limited Search (DLS)

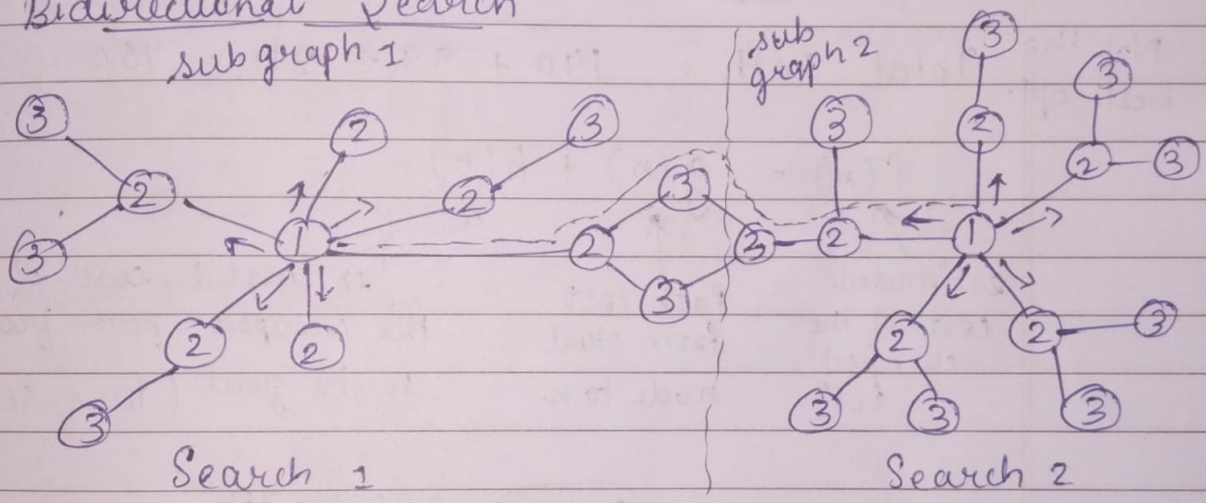
=) DLS is DFS with a depth limit L

=) Nodes at depth limit are treated as leaves

→ depth limit ? (how to decide)
→ Performance ?
Incomplete if $L < d$
Non Optimal if $L > d$
Time & Space <= DFS

★ ID - DFS (Iterative Deepening DFS)
Same as BFS

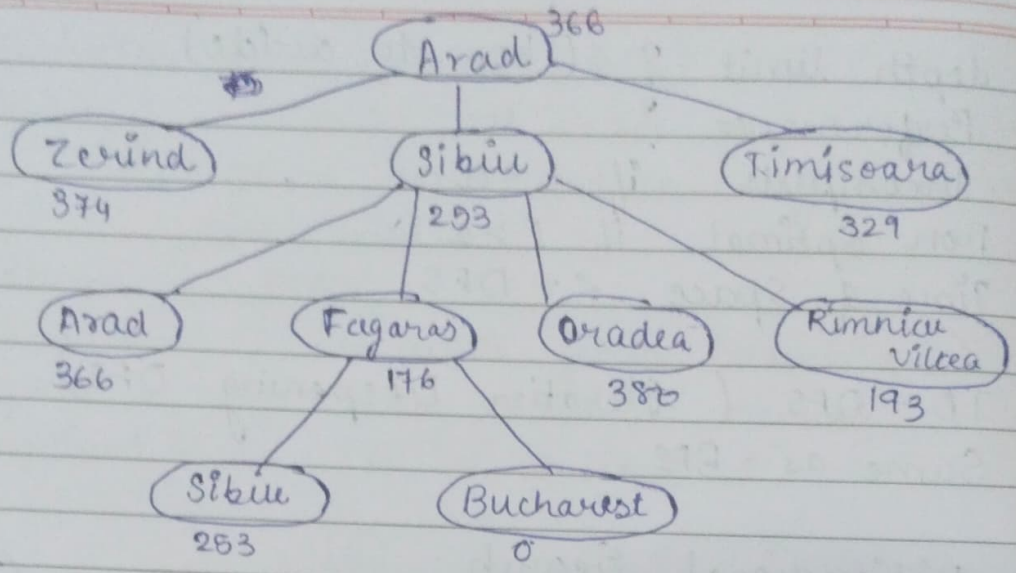
★ Bidirectional Search



⇒ It performs two search simultaneously one from the initial state to the goal state, the other from goal to initial state
⇒ It stops searching when either search reaches a state i.e in middle of the other state.
⇒ It provides complete & optimal sol^n if both searches are BFS.

★ Informed Search strategies
Eg: $h(n)$ heuristic
$f(n) = h(n)$
Greedy Best - First Search

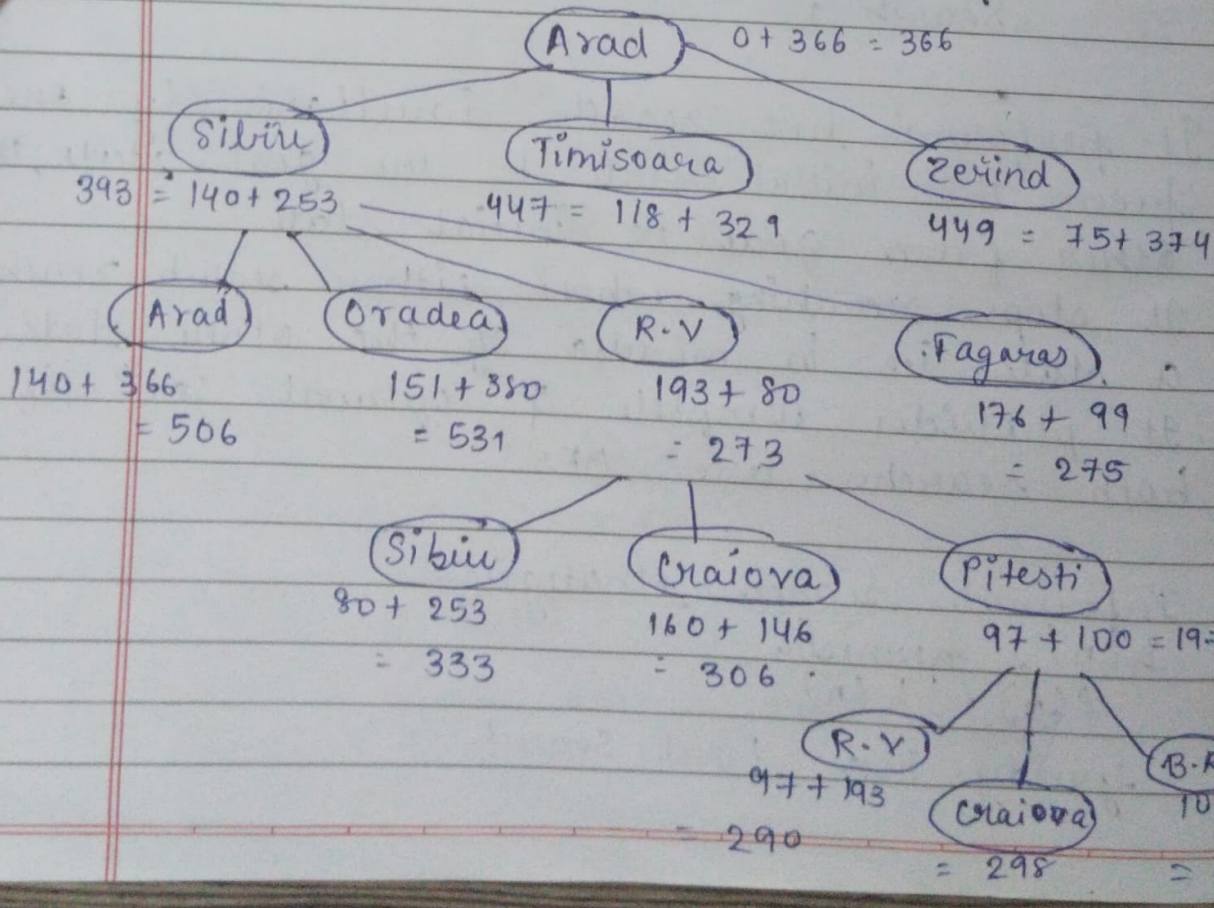
Eg:



Not the best opt. Total Path = 140 + 99 + 211 = 450

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

- $f(n)$: estimated cost of the cheapest soln n
- $g(n)$: Path cost from start node to n
- $h(n)$: estimated cost of the cheapest path from n to goal (here, SLD)

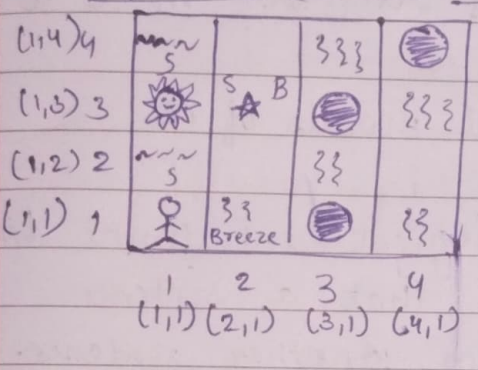


Total Path cost = 418 (140 + 80 + 97 + 101)

Informed Search (heuristic search)
- Greedy Best First search ($f(n) = h(n)$)
- A* - search (A* - algo)

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

- Wumpus World Problem



Pit
Wumpus
- Treasure
Agent
- Breeze
- Stench

~~PEAS~~ Solution :-

1) (1,1) → (2,1)
2) (2,1) → (1,1)
3) (1,1) → (1,2)
4) (1,2) → (2,2)

Step :- 1 Move to block [2,1] where agent smells breeze indicating adjacent block might have pit.

Step:2 Agent moves back to [1,1]

Step:3 Agent moves to [1,2] where agent feels stench indicating adjacent block might have wumpus

Step:4 Agent notices the difference in blocks [2,1] and [1,2] and notices block [2,2] is safe.

Step:5 Agent has now two opts either [3,2] or [2,3]

★ Logic :-
Syntax - (well formed sentence)
Semantics - meaning of sentences
model - possible world (logically mathematical abstraction)
logical entailment -

e.g If a sentence α is true in model m we say that m satisfies α or m is a model of α
$M(\alpha)$ ← set of all models of α

- Logical entailment :- Logical entailment b/n sentences is the idea that a sentence follows logically from another sentence.

e.g $\alpha \models \beta$
Sentence α entails the sentence β.

- Logical Inference is the process of deriving new sentences from the old one.

=> Sound Inferenie algorithms derive only sentences that are entailed, complete algorithms derive all sentences that are entailed.

★ Propositional logic :-