

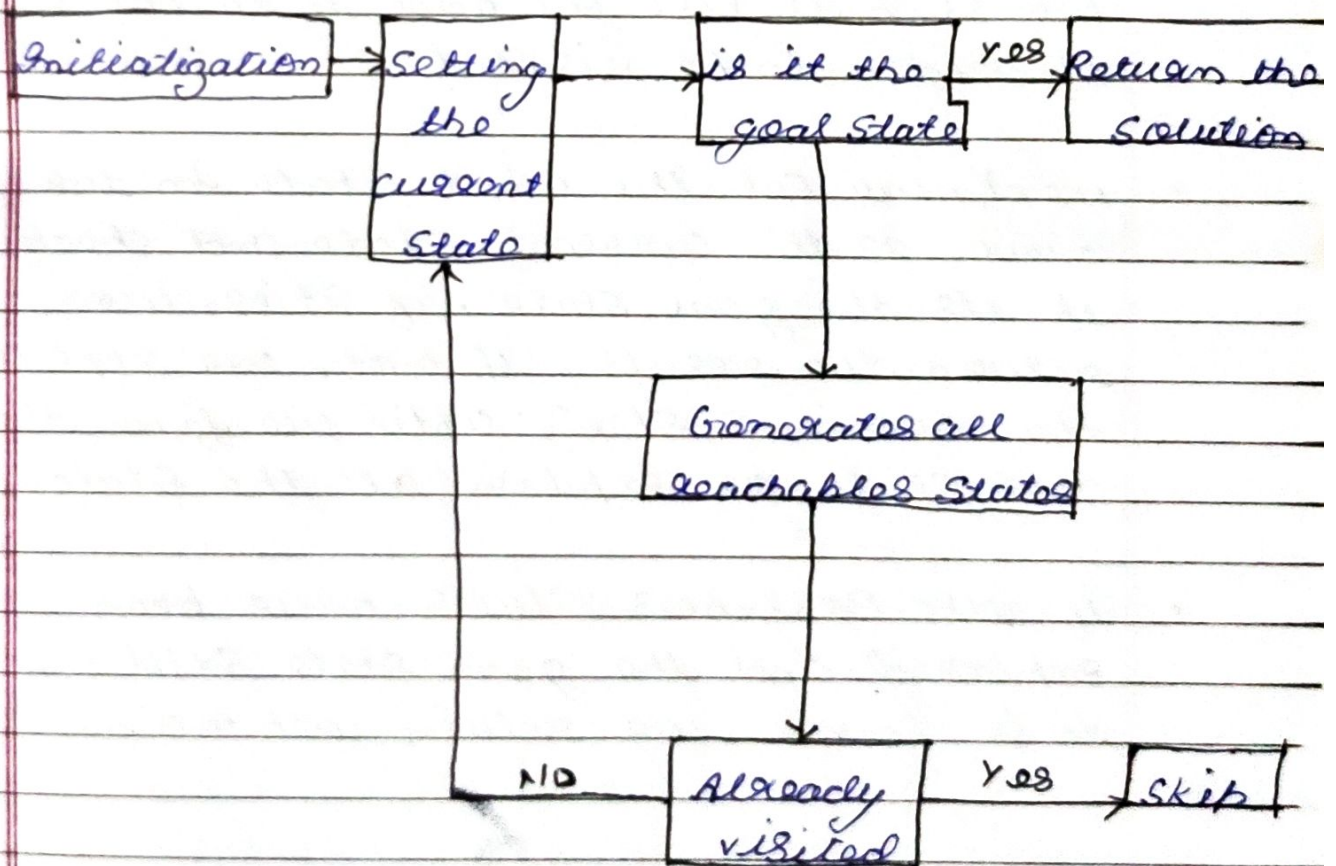


Date ___/___/___

- To begin the search process, we set the current state to the initial state.
- we then check if the current state is the goal state. if it is, we terminate the algorithm and return the result.
- If the current state is not the goal state, we generate the set of possible successor state that can be reached from the current state.
- For each successor state, we check if it has already been visited. if it has we skip it, else we add it to the queue of state to be visited.
- Next, we set the next state in the queue as the current state and check if it's the goal state. if it is, we return the result, if not, we repeat the previous steps until we find the goal state, or explore all the states.
- if all possible states have been explored and the goal state still needs to be found, we return with no solution.



- completeness.
if a solution exists, State Space Search will find a solution.
- optimality.
Search through a State Space result in optimal solution.
- uninformed and informed Search.
State space search in artificial intelligence can be classified as uninformed if it provides additional information about the problem.





Introduction of ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) refers to the development of computer system of performs tasks that required human intelligence. AI aids, in processing amounts of data identifying patterns and making decisions based on the collected information. This can be achieved through techniques like machine learning, natural language processing, computer vision and Robotics.

What is State Space Search?

State Space Search is a Problem-Solving technique used in Artificial Intelligence (AI) to find the solution path from the initial state to the goal state by exploring the various state. The State Space Search approach search through all possible states of a problem to find a solution.

Features of State Space Search.

State Space Search has several features that make it an effective problem-solving technique in Artificial Intelligence.

- Exhaustiveness:-

State Space Search explores all possible state of a problem to find a solution.



Q.

1		2
3	4	5
6	7	7

1	4	2
3	7	5
6		7

current state

Target state

Solution

1		2
3	4	5
6	7	7

Slide 1 right

	1	2
3	4	5
6	7	7

Slide 4 up

1	4	2
3		5
6	7	7

Slide 2 left

1	2	
3	4	5
6	7	7

Slide 7 up

1	4	2
3	7	5
6		7

reach the Target state.



Date ___/___/___

Informal Search

Known as It is also known as Heuristic Search

using knowledge It uses knowledge for the Searching Process

Perform once It finds a solution more quickly

completion It may or may not be complete

Cost factor cost is low

Time It consumes less time because of quick Searching

Direction There is a direction given about the solution

Imple - mental It is less lengthy while implemented

Ex of Algorithm

- Greedy Search
- A* Search
- A⁰* Search
- Hill climbing Algorithm

Uninformal Search

It is also known as Blind Search

It doesn't use knowledge for the Searching Process.

It finds solution slow as compared to an informal Search.

It is always complete

cost is high

It consumes more time because of slow Searching.

NO suggestion is given regarding the solution in it.

It is more lengthy while implemented

• Depth first Search

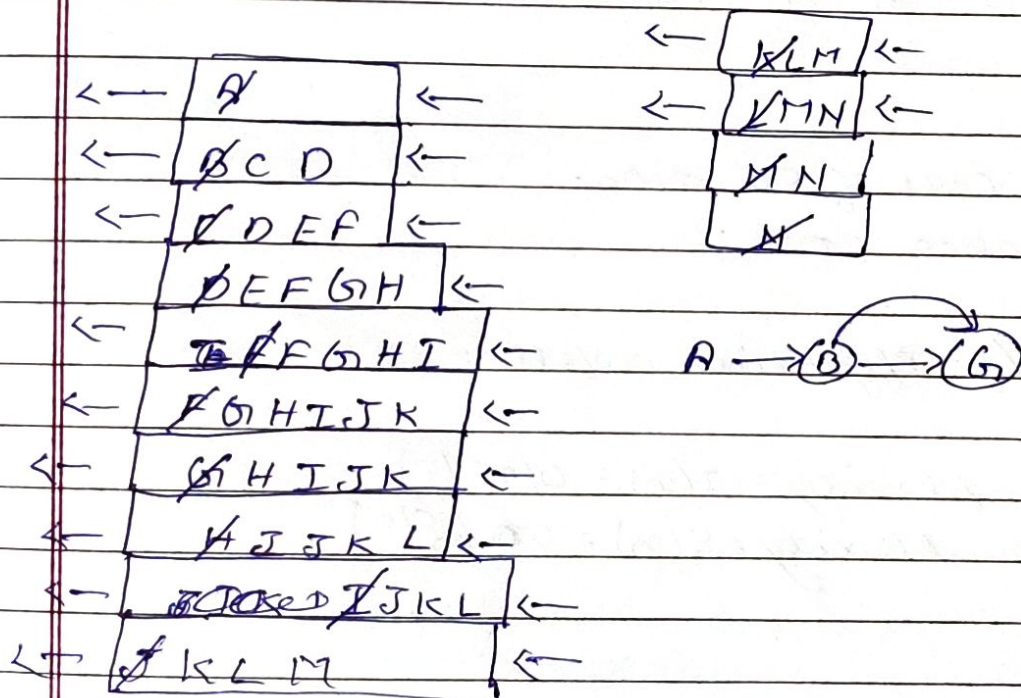
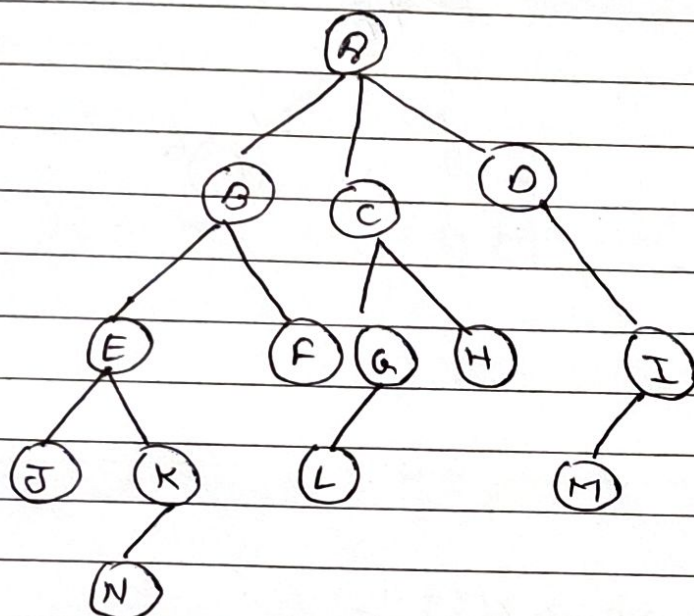
• Breadth first Search

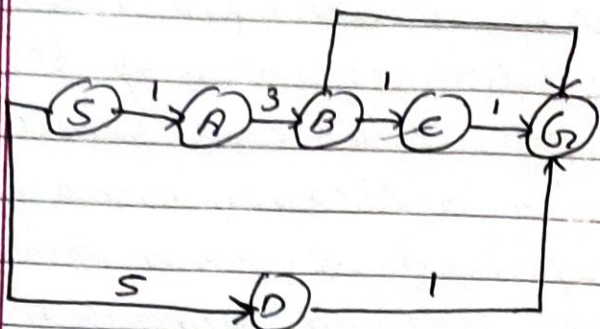
• Branch and Bound.



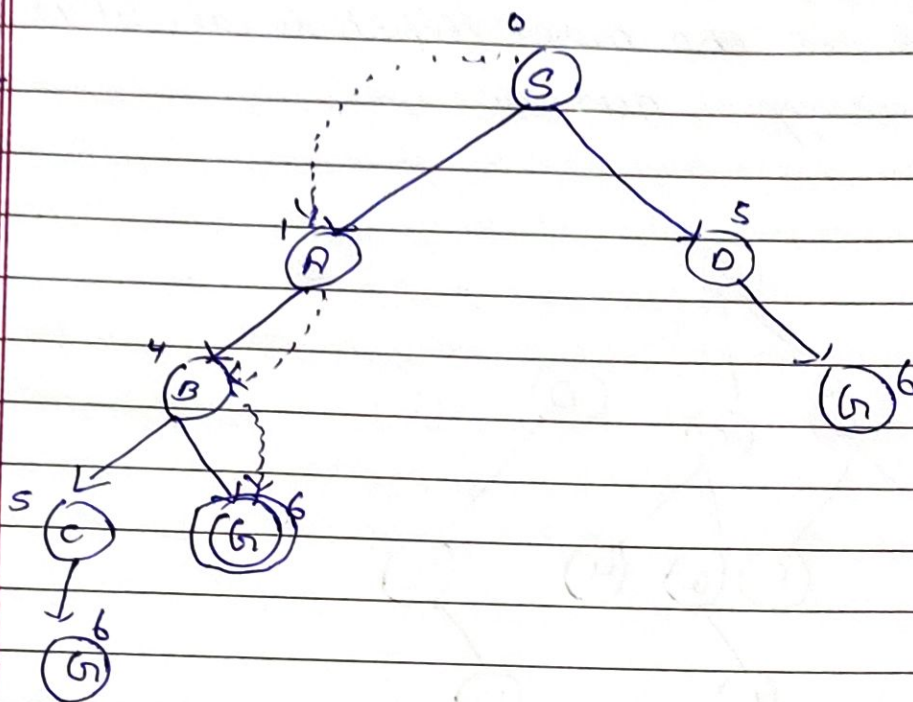
Breadth first Search $\Rightarrow$ Breadth-first Search (BFS) is an algorithm for traversing or searching tree or graph data structure. It starts at the tree root (or some arbitrary node for a graph, some times referred to as a search key), and explores all of the neighbour nodes at the present depth prior to moving on to the nodes at the next depth level. It is implemented using a queue.

Q.





Solution



Path $\Rightarrow S \rightarrow A \rightarrow B \rightarrow G$

Cost $\Rightarrow 5$

Let $C = \text{Cost of Solution}$.

$E = \text{edge cost}$.

Then $C/E = \text{effective depth}$.

Time complexity $= T(n) = O(n^{C/E})$

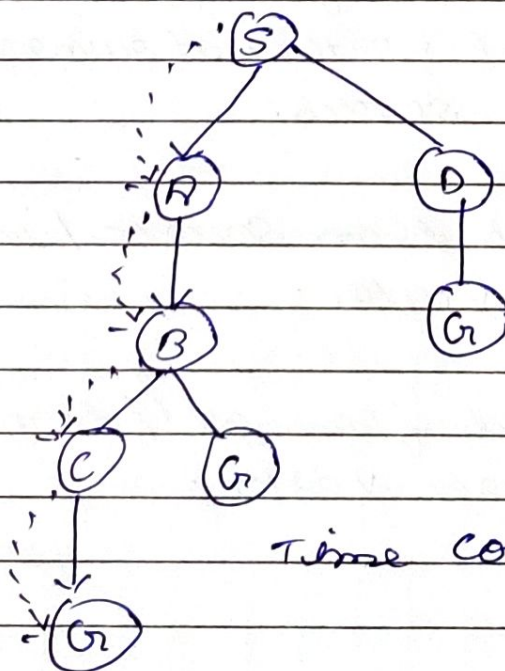
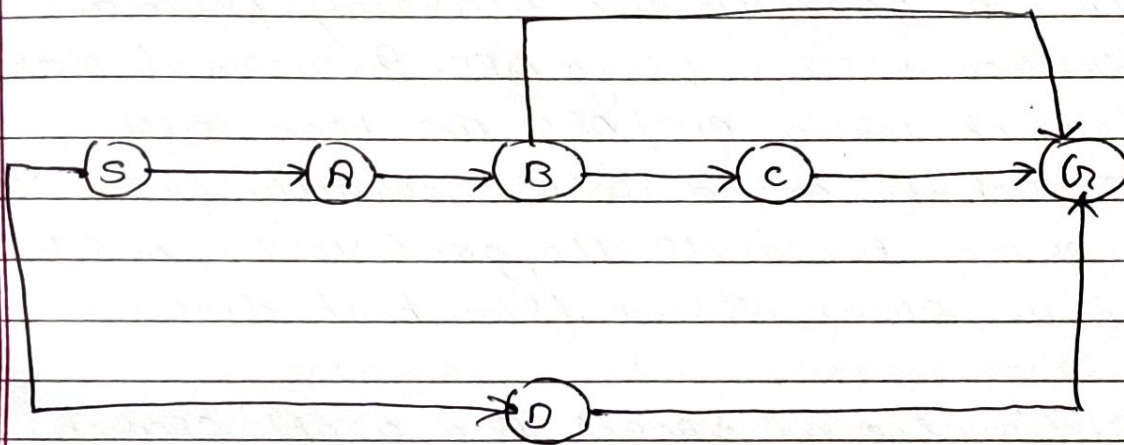
Space complexity $= S(n) = O(n^{C/E})$.



DFS

Depth-first Search (DFS) is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking. It uses last in - first out and hence it is implemented using a stack.

Q.



d = the depth of the search tree = the number of level of the search tree.

n^i = number of nodes in level.

Time complexity = $O(n^d)$

$S \rightarrow A \rightarrow B \rightarrow C \rightarrow G$



- Uniform Search Technique
- Stack (LIFO)
- Deepest Node
- Incomplete
- Non-optimal

Bidirectional Search.

Searching a graph is quite famous and have a lot of practical use. We have already discussed here how to search for a goal vertex starting from a source vertex using BFS. In normal graph search using BFS/DFS we begin our search in one direction usually from source vertex towards the goal vertex, but what if we start search from both direction.

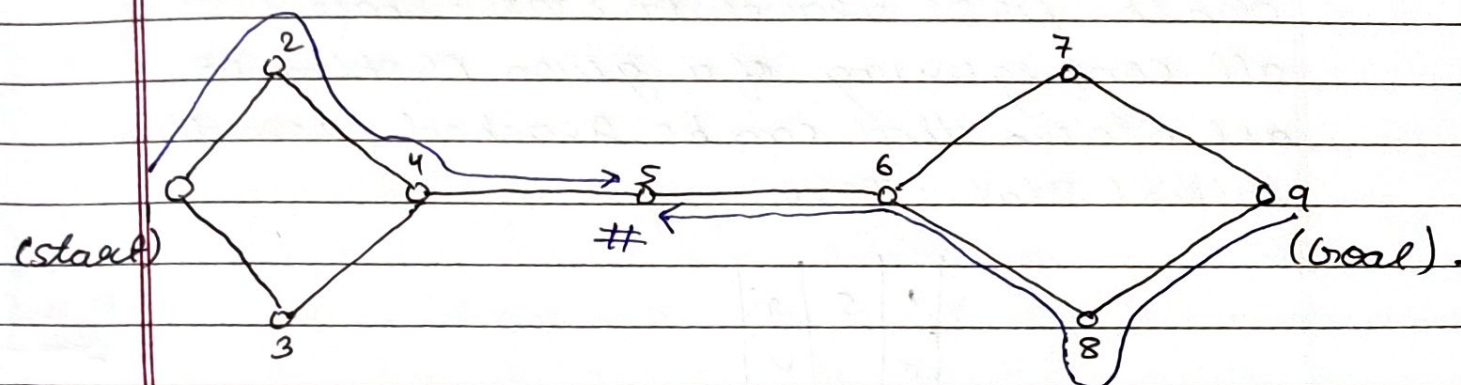
Bidirectional Search is a graph search algorithm which find smallest path from source to goal vertex. it runs two simultaneous search.

1. Forward Search from source/initial vertex towards goal vertex.
2. Backward Search from goal/target vertex towards source vertex.



Bidirectional Search.

- Two simultaneous search from an initial node to goal and backward from goal to initial stopping when two meet.
- Time complexity: $2(b^{d/2})$
- complete in breadth first search
- Not in depth first search.



8-Puzzle Problem without Heuristic.

In this puzzle solution of the 8 Puzzle problem is discussed.

Given a 3×3 board with 8 tiles (every tile has one number from 1 to 8) and one empty space. The observation is to place the numbers on tiles to match the final configuration using the empty space. we can slide four adjacent (left, right, above and below) tiles into the empty space.



Date ___/___/___

2. BFS (Brute-force) $\Rightarrow$ we can perform a Breadth-first search on the State Space Tree. This always finds a goal state nearest to the root. But no matter what the initial state is, the algorithm attempts the same sequence of moves like DFS.

1. DFS (Brute-force) $\Rightarrow$ we can perform a depth-first search on State-space (set of all configurations of a given problem i.e. all states that can be reached from the initial state) Tree.





In this solution, successive move can take us away from the goal rather than bringing us closer. The search of state-space tree follows the leftmost path from the root regardless of the initial state. An answer node may never be found in this approach.

- Blind Search (uninformed)
- BFS
- $O(b^d)$
- 4 moves (up, down, left, Right)

3. Branch and Bound $\Rightarrow$ The search for an answer node can often be speeded by using an "intelligent" pruning function, also called an approximate cost function to avoid searching in sub-trees that do not contain an answer node. It is similar to the backtracking technique but uses a BFS like search.

	2	3
1	8	4
7	6	5

1. live node is a node that has been generated but whose children have not yet been generated.

2		3
1	8	4
7	6	5

2. E-node is a live node whose children are currently being explored.

1	2	3
7	8	4
6		5

3. dead node is a generated node that is not to be expanded ☐ any further.



18.

Heuristics.

A heuristic is a technique that is used to solve a problem faster than the classic methods. These techniques are used to find the approximate solution of a problem when classical methods do not. Heuristics are said to be the problem solving techniques that result in practical and quick solution.

Direct Heuristics search techniques in AI.

It includes Blind Search, Uniformed Search, and Blind control Strategy. These search techniques are not always possible as they require much memory and time. These techniques search the complete space for a solution and use the arbitrary ordering of operations.

Weak Heuristics search techniques in AI.

It includes Informed Search, Heuristic Search and Heuristic control Strategy. These techniques are helpful when they are applied properly to the right type of tasks. They usually require domain specific information.

The example of weak Heuristic search techniques includes Best first Search

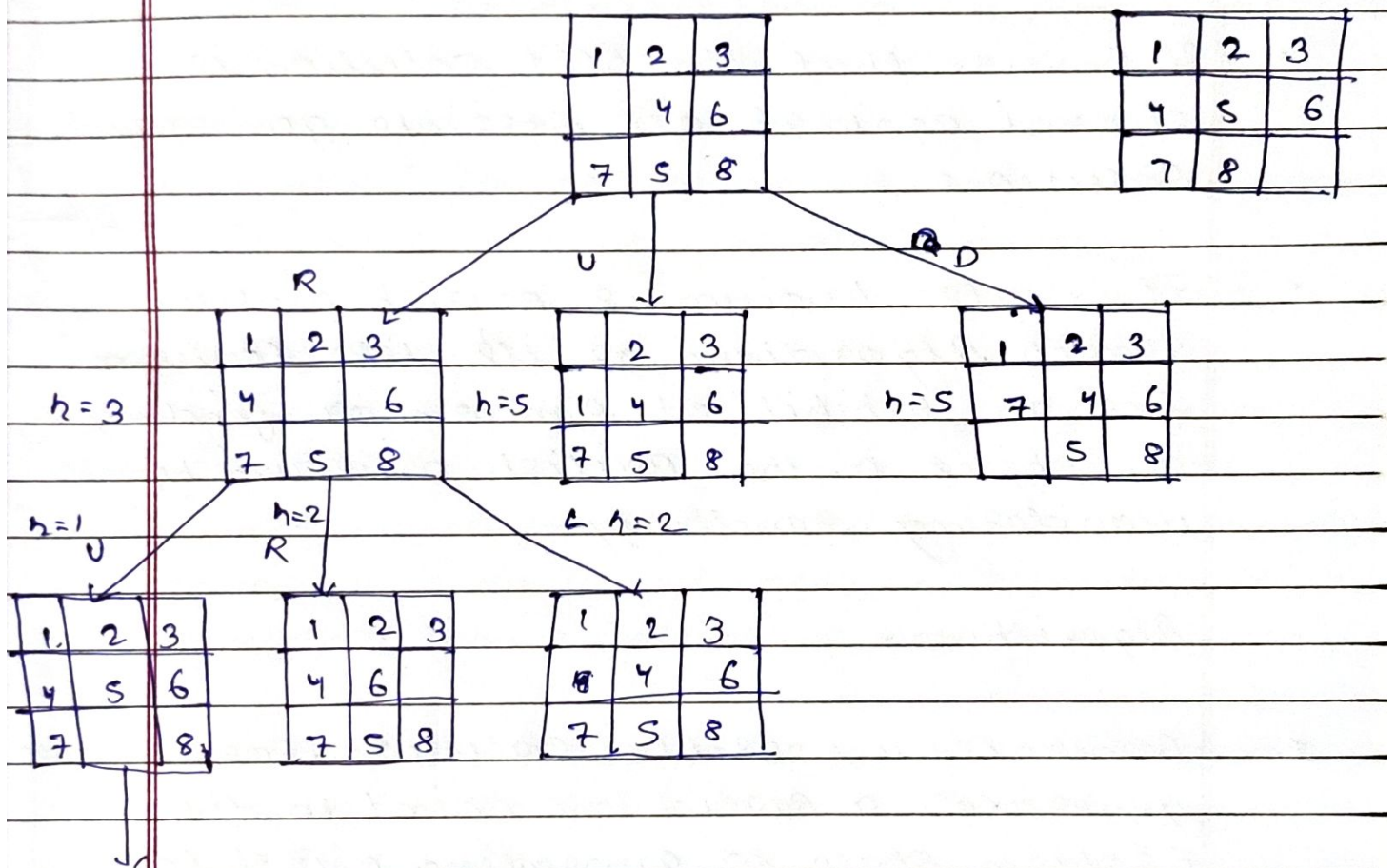
Date ___/___/___



(BFS) and A*.

- Bidirectional Search
- A* Search.
- Simulated Annealing
- Hill Climbing
- Best first Search
- Beam Search.

Q. How to solve 8-Puzzle Problem with Heuristic (Informed Search).



$$\text{Time} = O(b^d)$$

1	2	3
4	5	6
7	8	

Goal.



Generate and Test Algorithm.

- Generate and Test Search is a heuristic Search technique based on Depth First Search with backtracking which guarantees to find a solution if one exists systematically and there exists a solution.
- In this technique, all the solutions are generated and tested for the best solution.
- It ensures that the best solution is checked against all possible generated solutions.
- It is also known as British Museum Search Algorithm as it's like looking for an exhibit at random or finding an object in the British Museum by wandering randomly.

Algorithm.

1. Generates a possible solution. For example generates a particular point in the problem space or generating a path for a start state.
2. Test to see if this is an actual solution by comparing the chosen point or the end point of the chosen path to the set



of acceptable goal state.

3. If the solution is found quit. otherwise goto Step 1.

Properties of Good Generators.

- Complete
- Non-redundant
- Informed.

Best First Search (Informed, Heuristics)

The idea of Best first Search is to use an evaluation function to decide ~~out~~ which adjacent is most promising and then explore.

Analysis

- The worst-case time complexity for Best First Search is $O(n \cdot \log n)$ where n is the number of nodes. In the worst case, we may have to visit all nodes before we reach goal.
- The Performance of the algorithm depends on how well the cost or evaluation function is designed.

Special case of Best first Search.

1. Greedy Best first Search algorithm.
2. A* Search algorithm.



Date ___/___/___

→ let 'open' be a Priority queue containing
(b^d) initial state.

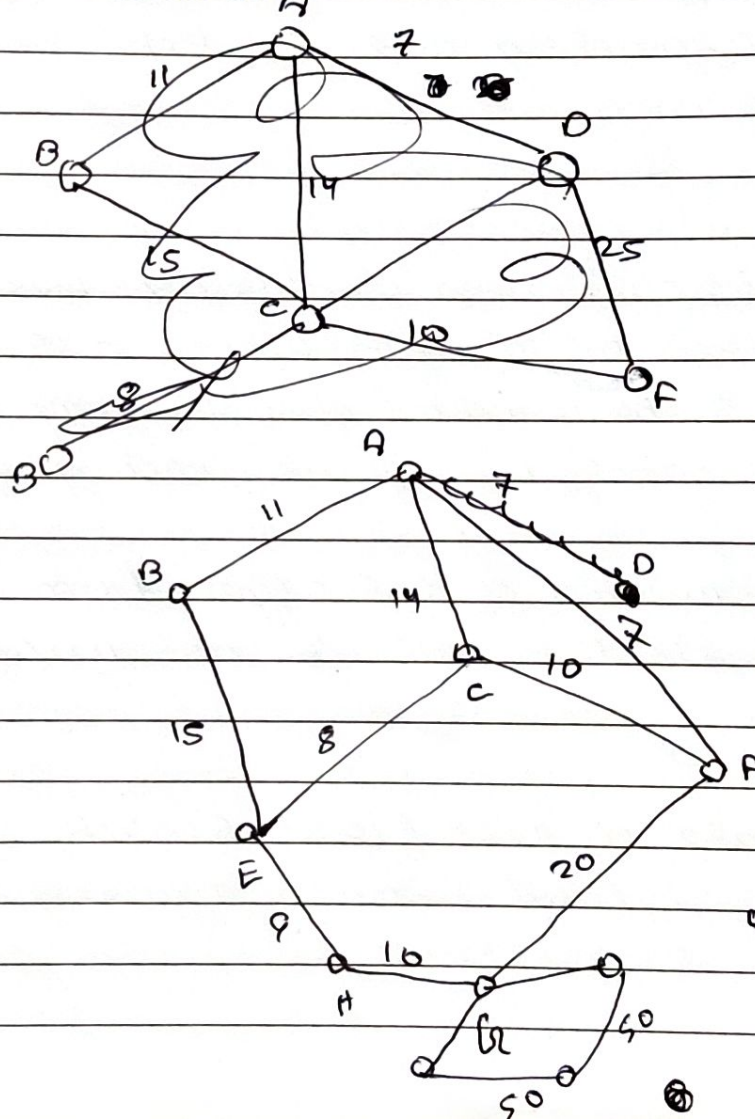
Loop.

if open is empty return failure

Node ← Remove - first (open)

if a goal Node is a goal
Then return the path from
initial to Node generate all
Successors of node and the
newly generated Node into
OPEN. according to their
f values END LOOP.

End loop.





straight line dist.

$$A \rightarrow G = 40$$

$$B \rightarrow G = 32$$

$$C \rightarrow G = 25$$

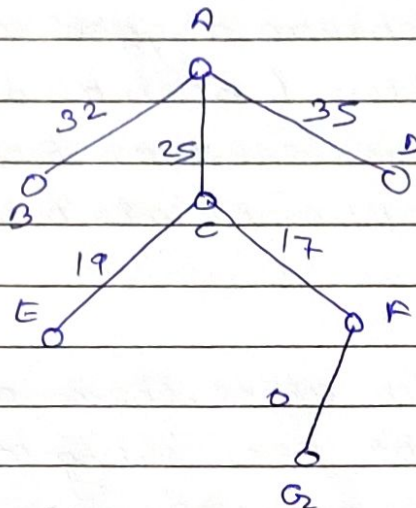
$$D \rightarrow G = 35$$

$$E \rightarrow G = 19$$

$$F \rightarrow G = 17$$

$$H \rightarrow G = 10$$

$$G \rightarrow G = 0$$



$$A \rightarrow C \rightarrow F \rightarrow G.$$



Beam Search Algo.

A heuristic techniques is a set of criteria for determining which of multiple options will be the most effective in achieving a particular goal.

This strategy increases the efficiency of a search process by surrendering claims of systematic and completeness of the best.

we can hope to achieve a good solution to difficult problem (such as the travelling salesman problem) in less than exponential time. if we use appropriate heuristics.

Beam Search

- A heuristic search algorithm a graph by extending the most promising node in a limited set is known as beam search.

- Beam search is a heuristic search technique that always expands the w number of the best node at each level.

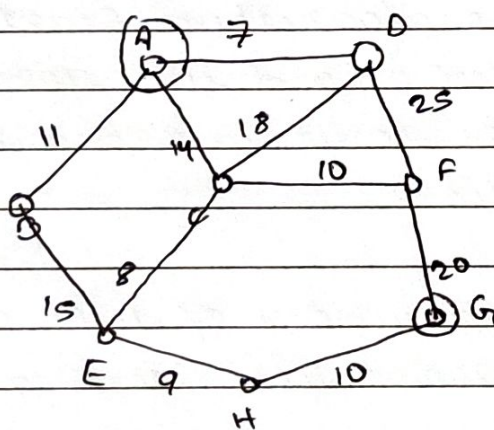
- It progresses level by level and moves downwards only from the best w nodes at each level.

- Beam search uses breadth-first search to build its search tree.



Date ___/___/___

- It generates all the successors of the current's level's state at each level of the tree.
- However, at each level, it only evaluates a w numbers of states.
- other nodes are not taken into account.
- the heuristic cost associated with the node is used to choose the best node.
- the width of the beam search is denoted by w .
- if B is a Branching factor, at every depth, there will always be $w \times B$ nodes under consideration but only w will be chosen.



Straight line

$$A \rightarrow G = 40$$

$$B \rightarrow G = 32$$

$$C \rightarrow G = 25$$

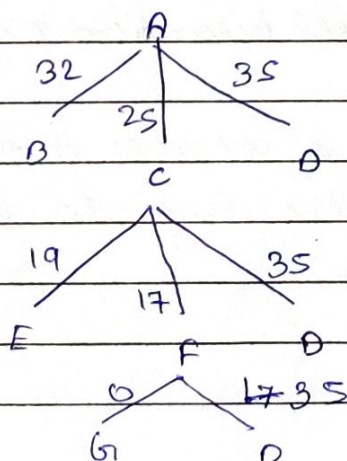
$$D \rightarrow G = 35$$

$$E \rightarrow G = 19$$

$$F \rightarrow G = 17$$

$$H \rightarrow G = 10$$

$$G \rightarrow G = 0$$



A C B D
C B D
E B D
G E B D



Simple Hill climbing Algorithm

- Take care of space complexity (constant).
- Beam width (B) is given.

Simple Hill climbing Algorithm

Hill climbing is a simple optimization algorithm used in Artificial Intelligence (AI) to find the best possible solution for a given problem. ~~where we where~~ the goal is to find the best solution from it belongs to the family of local search algorithms and is often used in optimization problem. ~~where~~ the goal is to find the best solution from a set of possible games.

- In Hill climbing, the algorithm starts with an initial solution and then iteratively makes small changes to it in order to improve the solution.

- There are several variations of Hill climbing including Steepest ascent Hill climbing.

Advantages of Hill climbing algorithm:

1. Hill climbing is a simple and intuitive algorithm that is easy to understand and implement.
2. It can be used in a wide variety of



optimization problem, including those with a large search space and complex constraints.

3. Hill climbing is often very efficient in finding local optima, making it a good choice for problems where a good solution is needed quickly.
4. The algorithm can be easily modified and extended to include additional heuristics or constraints.

Disadvantages of Hill climbing algorithm:-

1. Hill climbing can get stuck in local optima meaning that it may not find the global optimum of the problem.
2. The algorithm is sensitive to the choice of initial solution, and a poor initial solution may result in a poor final solution.
3. Hill climbing does not explore the search space very thoroughly, which can limit its ability to find the better solutions.
4. It may be less effective than other optimization algorithms, such as genetic algorithms or simulated annealing, for certain types of problems.



Features of Hill climbing.

1. Variation of generating and test algorithm.
 - Generates possible solutions.
 - Test to see if this is the expected solution.
 - If the solution has been found quite else go to step 1.
2. Uses the Greedy approach.

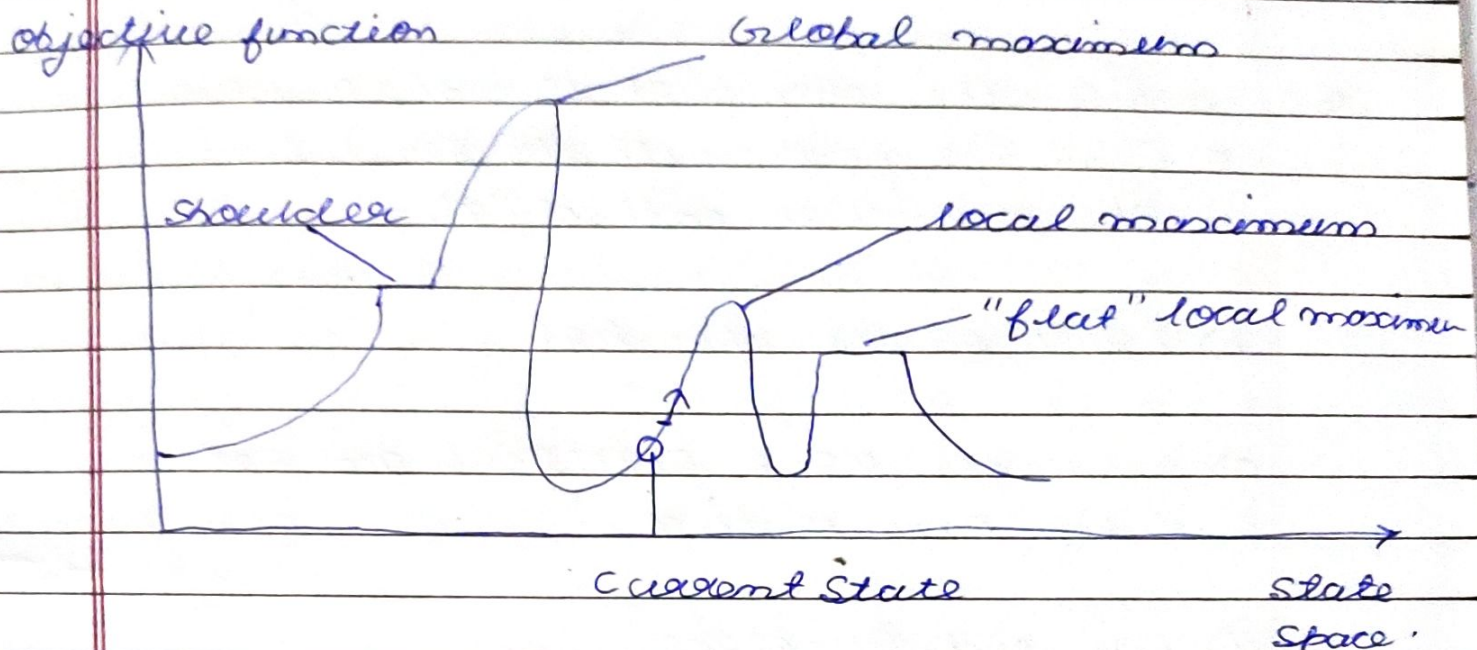
At any point in state space, the search moves in that direction only which optimizes the cost of function with the hope of finding the optimal solutions at the end.

Types of Hill climbing.

1. Simple Hill climbing $\Rightarrow$ It examines that neighbouring node one by one and selects the first neighbouring node which optimizes the current cost as the next node.
2. Steepest - Ascent Hill climbing $\Rightarrow$ It first examines all the neighbouring nodes and then selects the node that is closest to the selected state as of the next node.
3. Stochastic hill climbing $\Rightarrow$ It does not



examine all the neighbouring nodes before deciding which node to select. it just select a neighbouring node at random and decides whether to move to the neighbor or to examine another.



A* Search Algorithm.

A* Search Algorithm is one of the best and Popular technique used in Path-finding and graph traversal.

Why A* Search Algorithm?

A. Informally speaking A* Search algorithm, unlike other traversal. It has "heuristic", which it means is that is really a smart algorithm which separates it from the other conventional algorithm. This facts is cleared in detail in below sections. And it is also worth knowing that many games and web-based maps used this algorithm to find the shortest path very efficiently (approximation).



Admissibility of A* Algorithm.

The cost of reaching the goal state is assessed using an admissible heuristic is an informed search algorithm. However, if we need to discover a solution to the problem. The estimated cost must be lower than or equal to the true cost of reaching the goal state. The algorithm employs that allowable heuristic to determine the best-estimated route from the current node to the objective state.

The evaluation function in A* looks like.

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

$$f(n) = \text{Actual cost} + \text{estimated cost}.$$

n = Current node

$f(n)$ = Evaluation function.

$g(n)$ = The cost from the initial node to the current node.

$h(n)$ = Estimated cost from the current node to the goal state.

condition.

$h(n) \leq h^*(n) \therefore$ under estimation

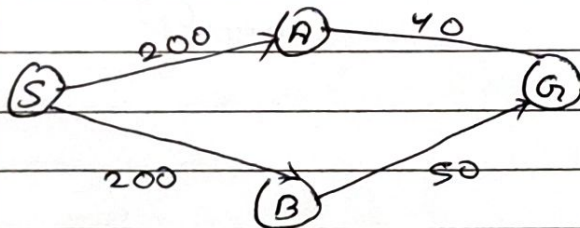
$h(n) \geq h^*(n) \therefore$ over estimation.



Key Points.

- The heuristic function $h(n)$ is admissible if $h(n)$ is never larger than $h^*(n)$ or if $h(n)$ is always less or equal to the true value.
- if A^* employs an admissible heuristic and $h(\text{goal}) = 0$ then we can argue that A^* is admissible.
- if the heuristic function is constantly tuned to be low with respect to the true cost i.e. $h(n) \leq h^*(n)$. Then you are going to get an optimal solution of 100%.

How to make A^* admissible



$$f(n) = g(n) + h(n) \leq h^*(n)$$

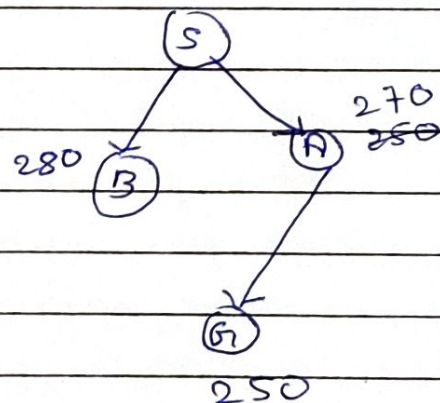
CASE 1. Overestimation.

$$\left. \begin{array}{ll} g(A) = 200 & h(A) = 80 \\ g(B) = 200 & h(B) = 70 \end{array} \right\} h^*$$

$$f(A) = 200 + 80 = 280 \times$$

$$f(B) = 200 + 70 = 270 \checkmark$$

$$f(G) = 250 + 0 = 250$$





Case II Under estimation-

$$h(A) = 30$$

$$h(B) = 20$$

$$f(A) = g(A) \geq h^*(A) \quad \text{Condition}$$

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

$$f(A) = 200 + 30 = 230 \times$$

$$f(B) = 200 + 20 = 220 \checkmark$$

$$f(G) = 200 + \overset{50}{20} = 250$$

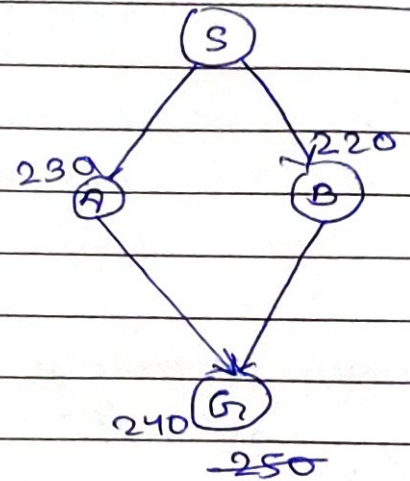
But we ~~can~~ ^{must} check the
Path is left.

$$f(G) = 200 + 40 = 240$$

which means.

$$S \rightarrow A \rightarrow G$$

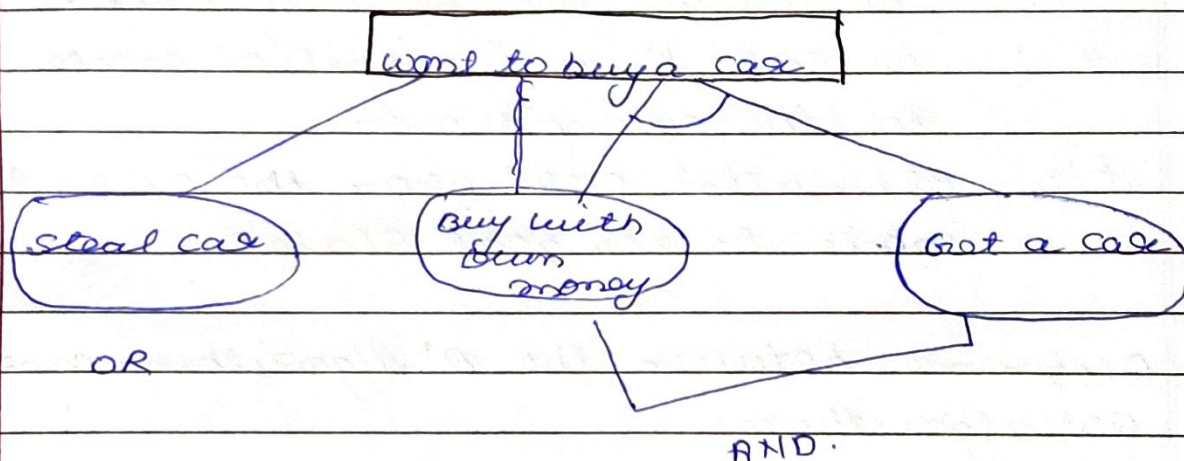
is the shortest Path.





AO* algorithm - Artificial Intelligence

Best-first Search is what the AO* algorithm does. The AO* method divides any given difficult problem into a smaller group of problems that are then resolved using the AND-OR graph concept. AND OR graphs are specialized graphs that are used in problem that can be divided into smaller problem. The AND side of the graph represents a set of tasks that must be completed to achieve the main goal, while the OR side of the graph represents different methods for accomplishing the same main goal.



In the above figure, the buying of a car may be broken down into smaller problems or tasks that can be accomplished to achieve the main goal. In the above figure, which is an example of a simple AND-OR graph. The other task is to either steal a car that will help us accomplish the main goal or use your own money to purchase a car that will accomplish the main goal. The AND symbol is used to indicate



The AND Part of the graphs, which refers to the need that all sub problems containing the AND to be resolved before the processing node or issue may be finished.

Working of AO* algorithm.

The evaluation function in AO* looks like this:

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

$f(n)$ = Actual cost + Estimated cost.

$f(n)$ = The actual cost of traversal.

$g(n)$ = The cost from the initial node to the current node.

$h(n)$ = Estimated cost from the current node to the goal state.

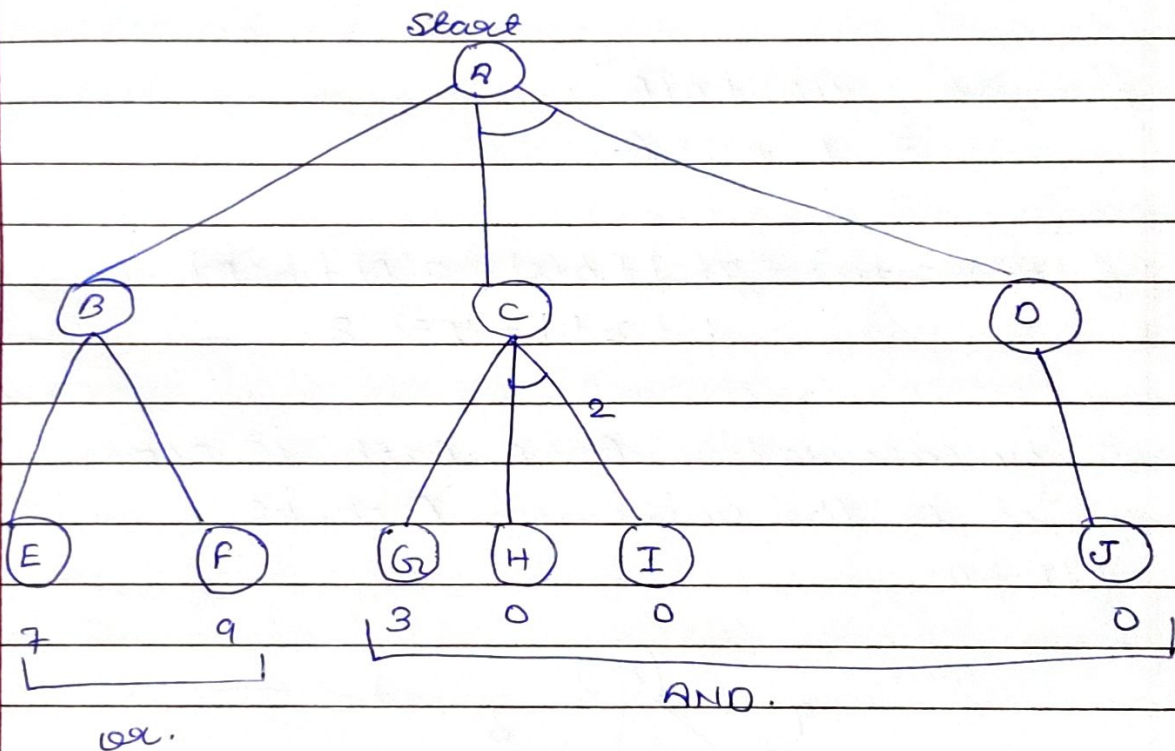
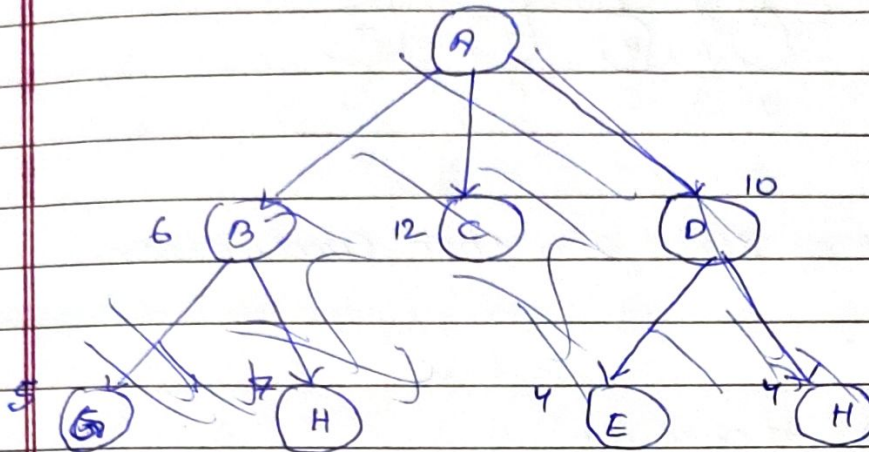
Difference between the A* Algorithm and AO* algorithm.

- A* algorithm and AO* algorithm works on the best first search.
- They are both informed search and works on given heuristic value.
- A* always gives the optimal solution but AO* doesn't guarantee to give the optimal solution.



When compared to the A^* algorithm, the AO^* algorithm uses less memory.

opposite to the A^* algorithm, the AO^* algorithm cannot go into endless loop.



Here in the above example below the node which is given in the heuristic value i.e $h(n)$. Edge length is considered as 1.



Introduction to Game Playing.

- Game Playing is an important domain of artificial intelligence. Game don't.

- The only knowledge we need to proceed in the rules, legal moves and the condition of winning or losing the game.

- Both Players try to win the game.

- So, both of them try to make the best move at each turn.

- Searching techniques like BFS (Breadth first Search) are not accurate for this as the branching factor is very high.

- So, Searching will take a lot of time.

we need other Search Procedures that improve.

- Generate Procedures so that only good moves are generated.

- Test Procedure so that best move can be explored first.

Ex:

One of the earliest examples of successful game playing AI is the chess program Deep Blue, developed by IBM, which defeated the world champion Garry Kasparov in 1997.

There are two main approaches to game playing in AI.

1. Rule-based systems use a set of fixed rules to play the game.
2. Machine Learning based systems: use algorithms to learn from experience and make decisions based on that experience.

Minimax algorithm uses two functions:-

MOVE GEN:- It generates all the possible moves that can be generated from the current position.

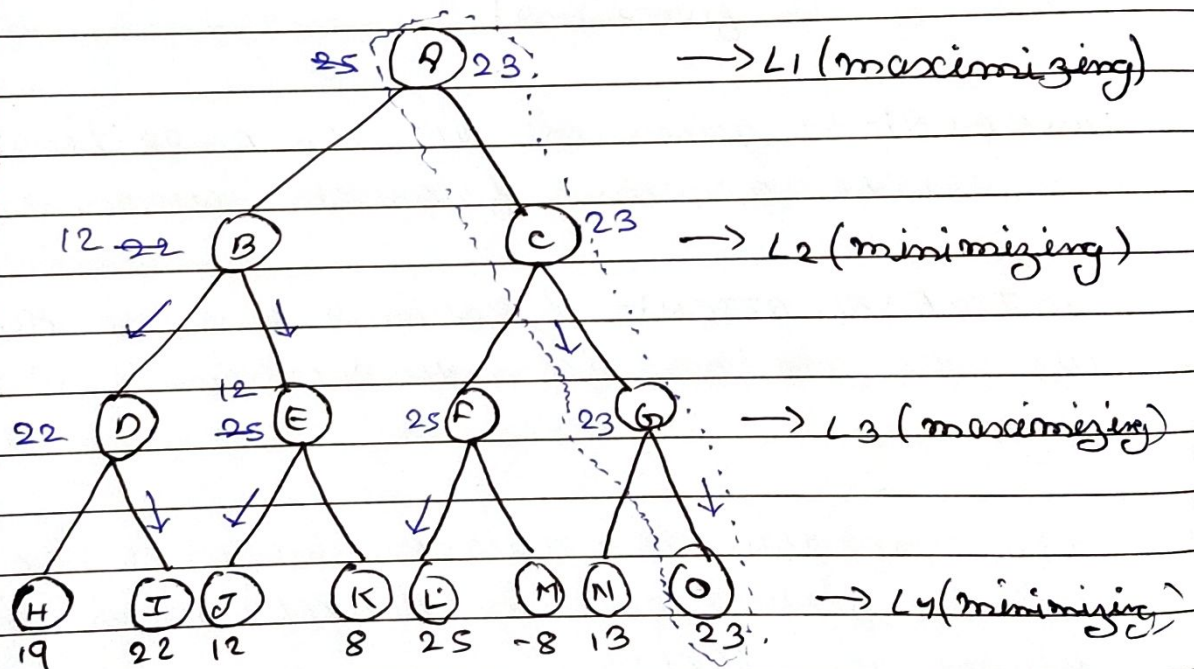
STATIC EVALUATION:- It returns a value depending upon the goalness from the viewpoint of two players.

- This algorithm is a two player game, so we call the first player as PLAYER1 and second player as PLAYER2.



Date ___/___/___

- The value of each node is backed-up from its children.
- For PLAYER1 the backed-up value is the maximum value of its children and for PLAYER2 the backed-up value is the minimum value of its children.
- It provides most promising move to PLAYER1, assuming that the PLAYER2 the backed-up value is the minimum value of its children.
- It provides most promising move to Player1, assuming that the PLAYER2 has make the best move.
- It is a recursive algorithm, as same procedure occurs at each level.





- 4 level are generated.
- The value of nodes H, I, J, K, L, M, N is provided by STATIC EVALUATION function.
- Level 3 is maximizing level, so all nodes of level 3 will take maximum value of their children.
- Level 2 is minimizing level, so all its nodes will take minimum values of their children.
- The value of A is 23.
- That means A should choose a move to win.

Advantages of Game PLAYING in Artificial Intelligence.

1. **Advancement of AI:** Game Playing has been a driving force behind the development of artificial intelligence and has led to the creation of new algorithms and techniques that can be applied to other areas of AI.
2. **Education and Training** $\Rightarrow$ Game Playing can be used to teach AI techniques and algorithms to students and professionals, as well as to provide for military and emergency response personnel.



Alpha-Beta Pruning (α - β).

- Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.
- As we have seen in the minimax search algorithm that the number of game state. It has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent but we can cut it to half. Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this is called pruning. This involves two threshold parameter Alpha and beta for further expansion. So it called alpha-beta pruning. It is also called alpha-beta algorithm.
- The two-Parameter can be defined as:
 - Alpha: The best (highest-value) choice we have found so far at any point along the path of maximizer. The initial value of α is $-\infty$
 - Beta: The best (lowest-value) choice we have found so far at any point along the path of minimizer. The initial value of β is $+\infty$

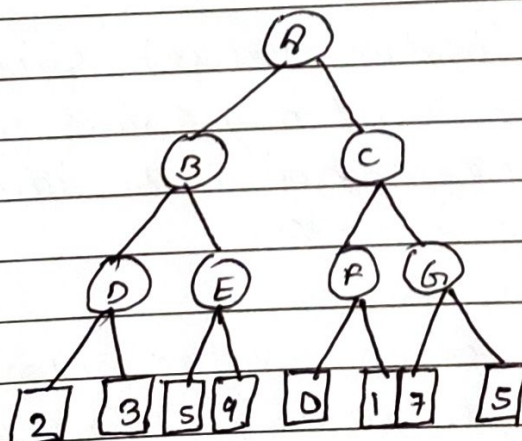


Date ___/___/___

Key Points about alpha-beta Pruning

- The max Player will only update the value of alpha.
- The min Player will only update the value of Beta.
- while backtracking the tree, the node values will be passed to upper node instead of values of alpha and beta.
- we will only pass the alpha, beta values to the child nodes.

STEP 1: At the first step, max player will start first move from node A where $\alpha = -\infty$ and $\beta = +\infty$, these value of alpha and beta passed down to node B where again $\alpha = -\infty$ and $\beta = +\infty$, and Node B passes the same value to its child D.



→ max

→ min

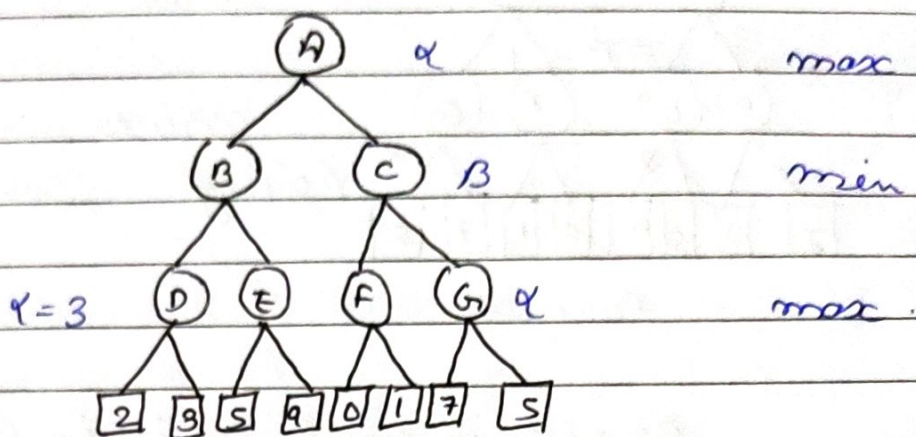
→ Max.

→ Terminal node

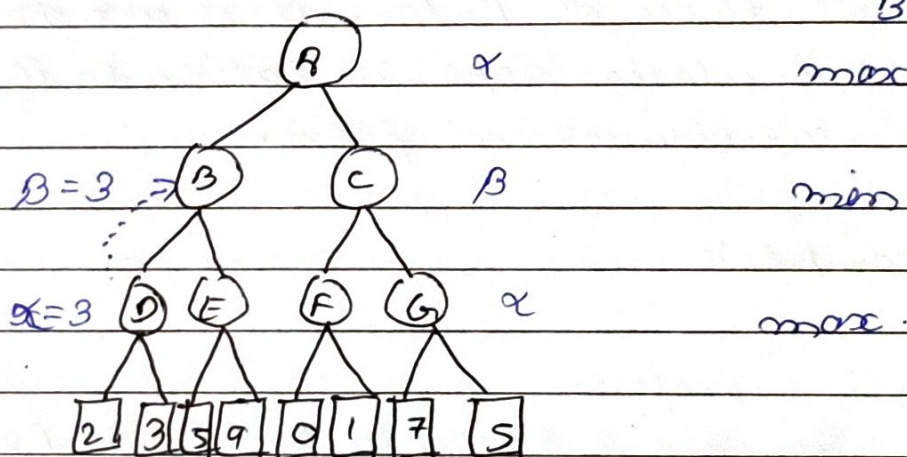


Date ___/___/___

Step 2: At Node D, the value of α will be calculated as its term for max. The value of α is compared with firstly 2 and then 3 the $\max(2, 3) = 3$ will be the value of α at node D and node value will also 3.



Step 3: Now algorithm back back to node B, whose the value of B will change as this is a turn of min, Now $B = \min$ will compare with the available subsequent nodes value i.e. $\min(\infty, 3) = 3$, hence at node B now $\alpha = -\infty$ and $B = 3$.

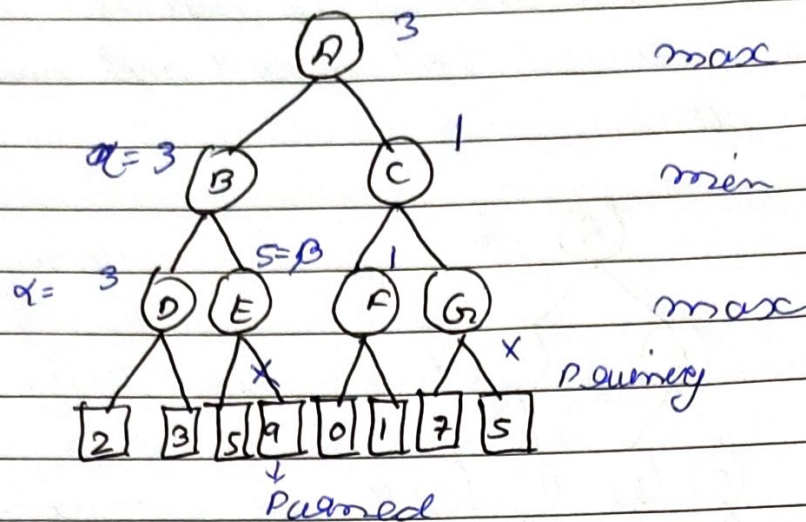


Step 4: At node E, min will take its turn, and the value of α Alpha will be compared with 5, so $\max(-\infty, 5) = 5$ hence $E\alpha = 5$ and $B = 3$, whose $\alpha \geq B$, so the right successor of E will be



Date ___/___/___

Pruned and algorithm will not traverse it, and the value at node E will be 5.



Propositional logic in Artificial Intelligence.

Propositional logic (PL) is the simplest form of logic where all the statements are made by Proposition. A Proposition is a declarative statement which is either true or false. It is a technique of knowledge representation in logical and mathematical form.

Example:-

- it is sunny.
- The sun rises from west (False Proposition)
- $3+3=7$ (false Proposition)
- 5 is a Prime number.



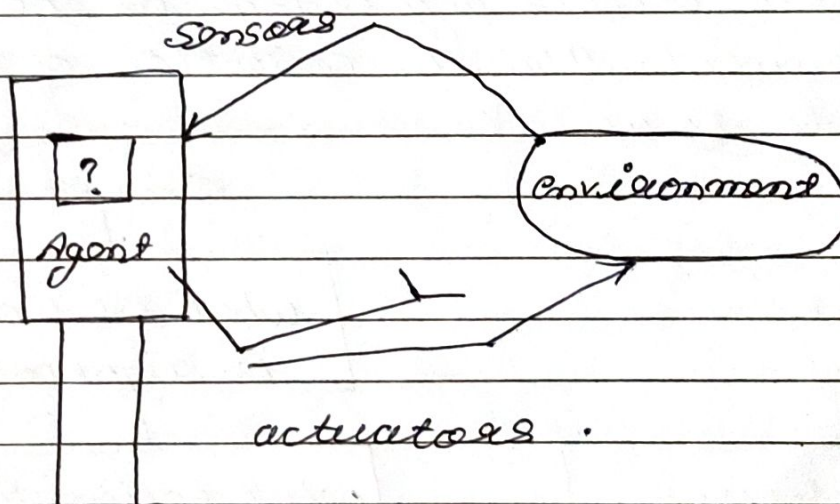
5. For Biconditional.

P	q	$P \leftrightarrow q$
True	True	True
True	False	False
False	True	False
False	False	True

Agents/Intelligent Agents.

In artificial intelligence, an agent is a computer program or system that is designed to perceive its environment, make decisions and take actions to achieve a specific goal or set of goals.

- Perceiving its environment through sensors and.
- Acting upon that environment through actuators.



Every agent can perceive its own actions (but not always the effects)



Date ___/___/___

Types of Agents.

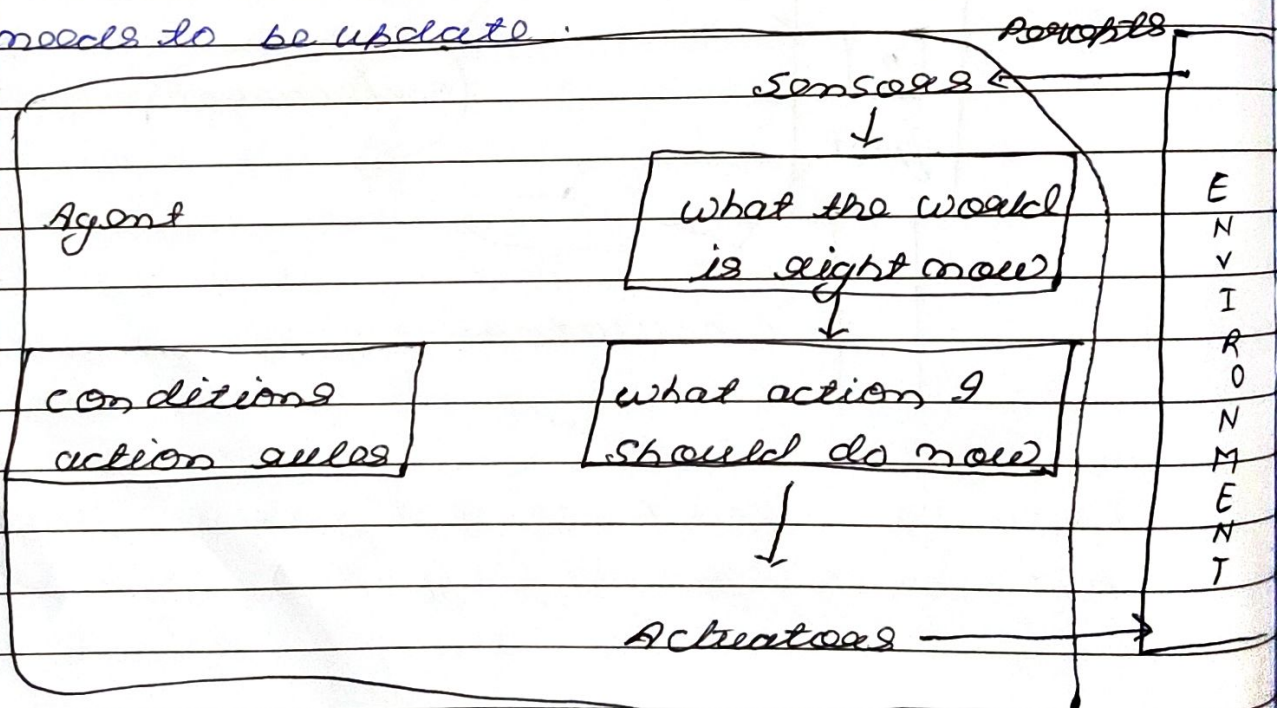
Agents can be grouped into five classes based on their degree to Perceive intelligence and capability.

Simple Reflex Agents.

Simple reflex agents ignore the rest of the Percept history and act only on the basis of the current Percept.

Problems with Simple reflex agents are.

- Very limited Intelligence.
- No-knowledge of non-Perceptual Part of State.
- usually too big to generate and store.
- if there occurs any change in the environment, then the collection of rules needs to be update.



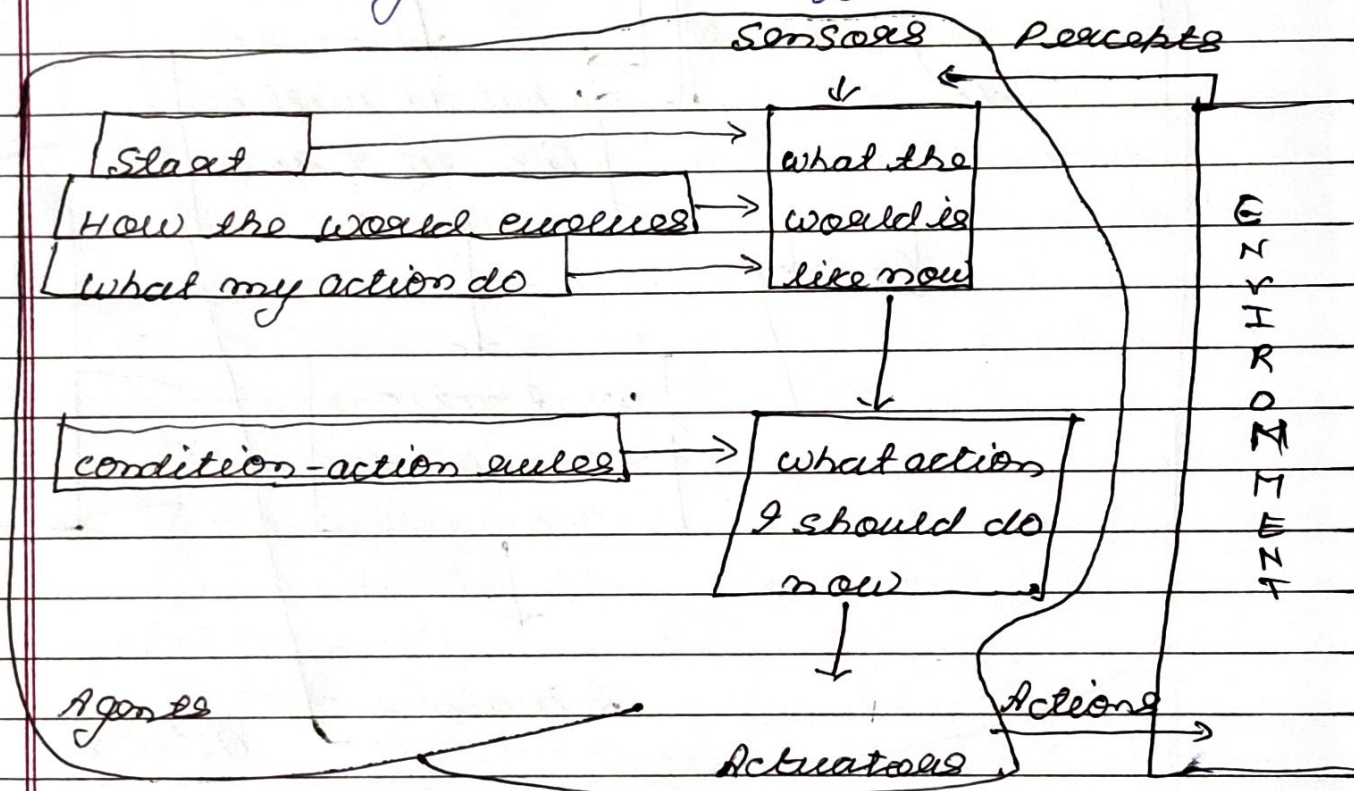


• Model-Based Reflex Agents.

It works by finding a rule whose conditions matches the current situation. A model-based agent can handle partially observable environments by the use of model about the world.

updating the state information about:-

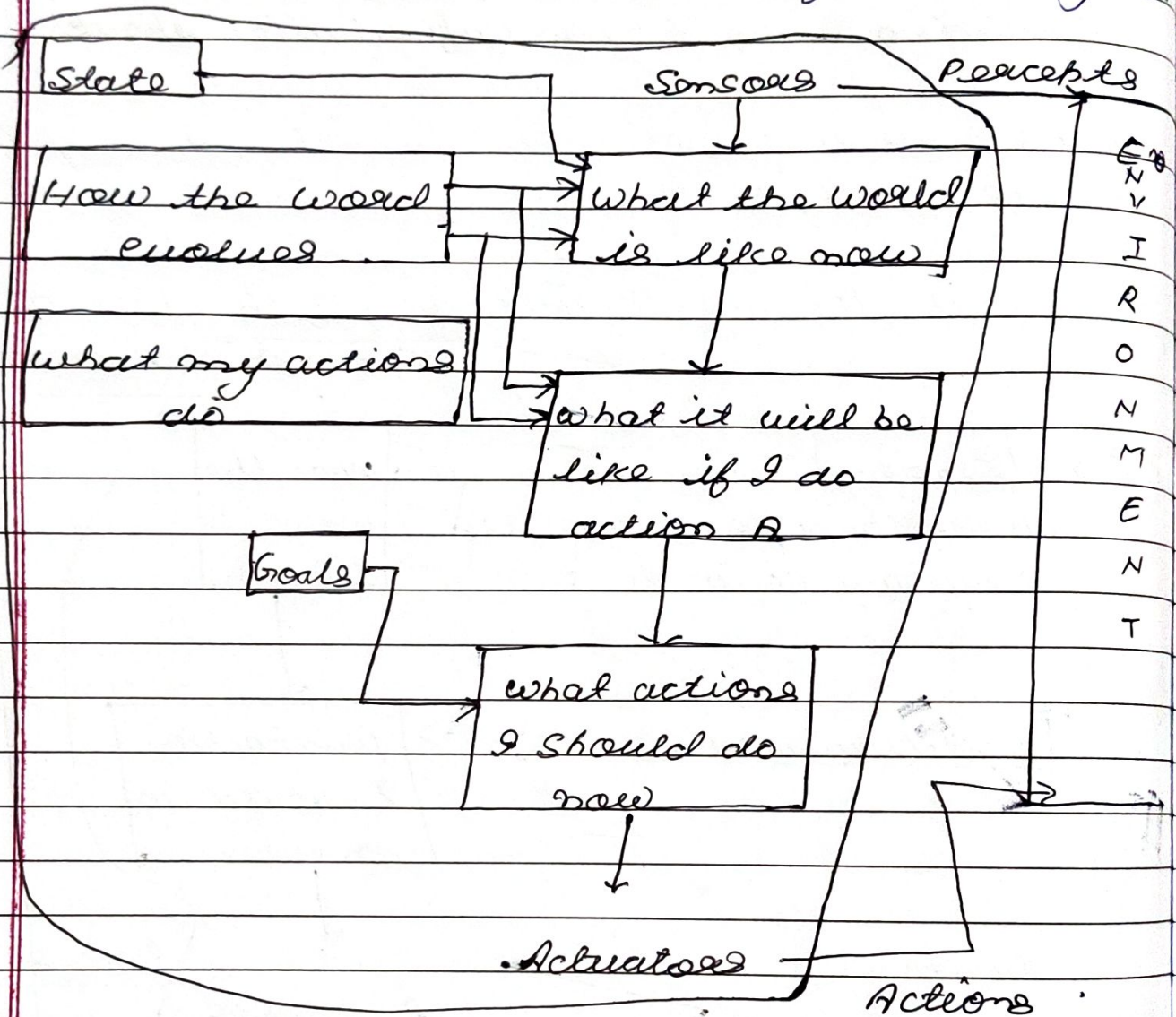
- ~~How~~ How the world evolves independently from the agent?
- How do the agent's actions effect the world?





Goal-Based Agents.

These kinds of agents take decisions based on how far they are currently from their goal (description of desirable situations.) Their every actions is intended to reduce their distance from the goal.

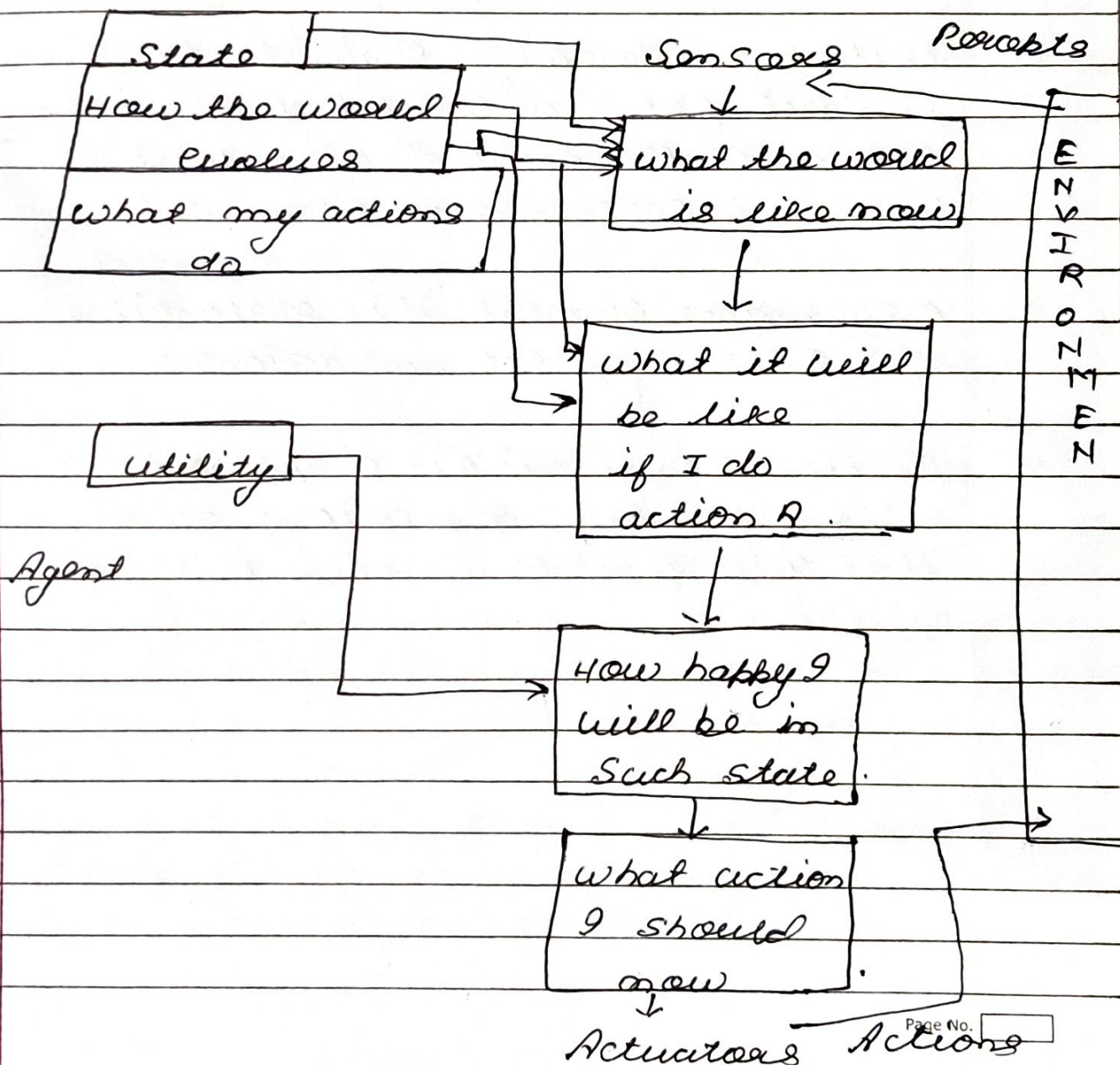


which makes these agents more flexible. They usually require search and planning. The goal based agent's behaviour can easily be changed.



Utility-Based Agents.

The agents which are developed having their end uses as building blocks are called utility based agents. When there are multiple possible alternatives, then to decide which one is best, utility based agents are used. They choose actions based on a Preference (Utility) for each state. Sometimes achieving the desired goal is not enough. We may look for a quicker.

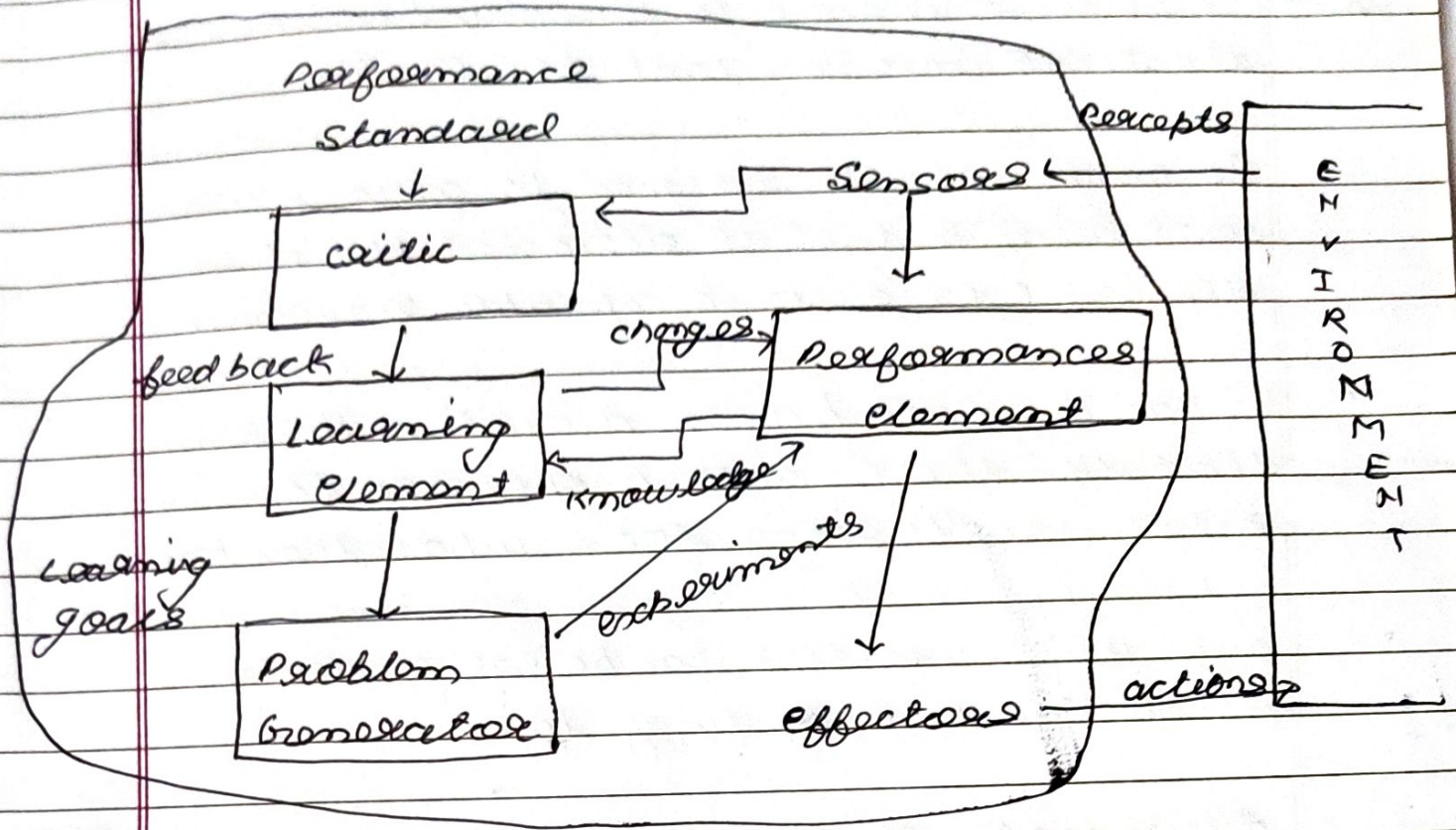




Learning Agent.

A learning agent in AI is the type of agent that can learn from its past experience or it has learning capabilities. It starts to act with basic knowledge and then is able to act and adapt automatically through learning.

1. **Learning element:** It is responsible for making improvements by learning from the environment.
2. **Critic:** The learning element takes from feedback from critics which describes how well the agent is doing with respect to fixed performance standard.
3. **Performance element:** It is responsible for selecting external actions.
4. **Problem Generator:** This component is responsible for suggesting action that will lead to new and informative experience.



FUZZY LOGIC.

- The term fuzzy refers to things that are not clear or are vague.
- Definition $\Rightarrow$ A graph that defines how each point in the input space is mapped to membership value between 0 and 1. Input space is often referred to as the universe of discourse or universal set (U), which contains all the possible elements of concern in each particular application.

There are largely three types of fuzzifiers:-

- Singleton fuzzifier.
- Gaussian fuzzifier.
- Trapezoidal or triangular fuzzifier.



Date ___/___/___

What is fuzzy control?

Ans. It is a technique to embody human-like thinkings into a control system.

- It may not be designed to give accurate reasoning but it is designed but it is designed to give acceptable reasoning.

- It can emulate human deductive thinking, that is, the process people use to infer conclusions from what they know.

- Any uncertainties can be easily dealt with the help of fuzzy logic.

Advantages of fuzzy logic system.

- This system can work with any type of inputs whether it is imprecise, distorted or noisy input information.

- The construction of fuzzy logic system is easy and understandable.

- Fuzzy logic comes with mathematical concepts of set theory and the reasoning of that is quite simple.

- It provides a very efficient solution to complex problems in all fields of life as it resembles human reasoning and decision-making.



Date ___/___/___

- The algorithm can be described with little data, so little memory is required.

Disadvantages of Fuzzy Logic Systems.

- Many researchers proposed different ways to solve a given problem through fuzzy logic which leads to ambiguity.

- Proof of its characteristics is difficult or impossible in most cases because every time we do not get a mathematical description of our approach.

- As fuzzy logic works on precise as well as imprecise data so most of the time accuracy is compromised.

Application.

- It is used in the aerospace field for attitude control of spacecraft and satellites.

- It has been used in the automotive system of speed control, traffic control.

- It is used for decision-making support system and personal evaluation in the large company business.

- It has application in the chemical industry for controlling the pH, drying, chemical distillation process.



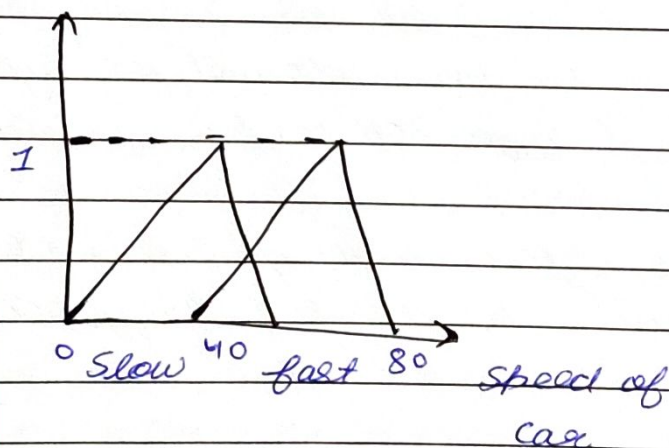
Fuzzy logic is used in Natural language processing and various intensive applications in Artificial Intelligence.

Fuzzy logic is extensively used in modern control system such as expert systems.

Represent uncertainty.

Represent will degree.

Represent the belongingness of a member of a crisp set to fuzzy set.





' check the degree of fastness.

0, if $\text{Speed}(x) \leq 40$

$\frac{\text{Speed}(x) - 40}{10}$, if $40 < \text{Speed}(x) < 50$

1, if $\text{Speed}(x) \geq 50$.

Q.

$$U = \{1, 2, 3, 4, 5\}$$

$$S = \{1, 2\}$$

$$= \{(1, 1), (2, 1), (3, 0), (4, 0), (5, 0)\}$$

Q.

$$x = 30$$

$$x = 60$$

$$30 \leq 40 \\ \Rightarrow (30, 0) \text{ A.}$$

Q. $x = 42$.

$$\frac{\text{Speed}(x) - 40}{10} = \frac{42 - 40}{10} = \frac{2}{10} = \frac{1}{5} = 0.2$$

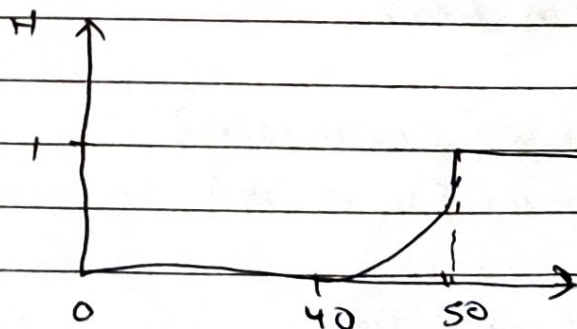
Q. $x = 45$.

$$60 \geq 40$$

$$(\cancel{40}, 1) \quad (60, 1) \text{ A.}$$

$$\frac{\text{Speed}(x) - 45}{10} = \frac{45 - 40}{10} = \frac{5}{10} = \frac{1}{2} = 0.5$$

graphical way representation.





Date ___/___/___

Operations in fuzzy logic

1. Union $\Rightarrow$ The operation is denoted by $(A \cup B)$. $A \cup B$ is the set of those elements which exist in two different sets A and B . This operation combines all the elements from both the sets and make a new set. It is called a logical OR operation.

$$A \cup B = \{x / x \in A \text{ or } x \in B\}$$

Example.

$$\text{Set } A = \{10, 11, 12, 13, 14, 15\}$$

$$\text{Set } B = \{11, 12, 13, 14, 15\}$$

$$A \cup B = \{10, 11, 12, 13, 14, 15\}$$

2. Intersection $\Rightarrow$ This operation is denoted by $(A \cap B)$, $A \cap B$ is the set of those elements which are common in both sets A and B . It is also called a logical operation.

$$A \cap B = \{x / x \in A \text{ and } x \in B\}$$

Example.

$$\text{Set } A = \{10, 11, 12, 13\}$$

$$\text{Set } B = \{11, 12, 14\}$$

$$A \cap B = \{11, 12\}$$



3. ~~Difference operation $\Rightarrow$ This operation is denoted by $(A-B)$. $A-B$ is the set of only those elements which exist only in set A but not in set B.~~

~~$$A-B = \{x / x \in A \text{ And } x \notin B\}$$~~

3. complement: $\mu_{\bar{A}}(x) = [1 - \mu_A(x)], x \in U.$

Q $A = \{(10, 0.2), (20, 0.4), (25, 0.7), (30, 0.9), (40, 1)\}$

$$\mu_{\bar{A}}(x) = (10, 0.8), (20, 0.6), (25, 0.3), (30, 0.1), (40, 0)\}$$

4. Bold union: $\mu_{A \oplus B} = \min[1, \mu_A(x) + \mu_B(x)]$

Q $A = \{(10, 0.2), (20, 0.4), (25, 0.7), (30, 0.9), (40, 1)\}$

$B = \{(10, 0.4), (20, 0.1), (25, 0.9), (30, 0.2), (40, 0.6)\}$

$$\mu_{A \oplus B} = \{(10, 0.6), (20, 0.5), (25, 1), (30, 1), (40, 1)\}$$

5. Bold Intersection: $\mu_{A \odot B}(x) = \max[0, \mu_A(x) + \mu_B(x) - 1]$

Q $A = \{(10, 0.2), (20, 0.4), (25, 0.7), (30, 0.9), (40, 1)\}$

$B = \{(10, 0.4), (20, 0.1), (25, 0.9), (30, 0.2), (40, 0.6)\}$

$$\mu_{A \odot B} x \Rightarrow \{(10, 0), (20, 0), (25, \overset{0.6}{\cancel{+0.6}}), (30, \overset{0.1}{\cancel{+0.1}}), (40, \overset{0.6}{\cancel{+0.6}})\}$$

6. Equality $A=B$ if $\mu_A(x) = \mu_B(x) \forall x \in S.$

Q $A = \{(10, 0.2), (20, 0.4), (25, 0.7), (30, 0.9), (40, 1)\}$

$B = \{(10, 0.4), (20, 0.1), (25, 0.9), (30, 0.2), (40, 0.6)\}$

$$\mu_A(x) =$$



Date ___/___/___

6. Product.

$$\mu_A \cdot \mu_B(x).$$

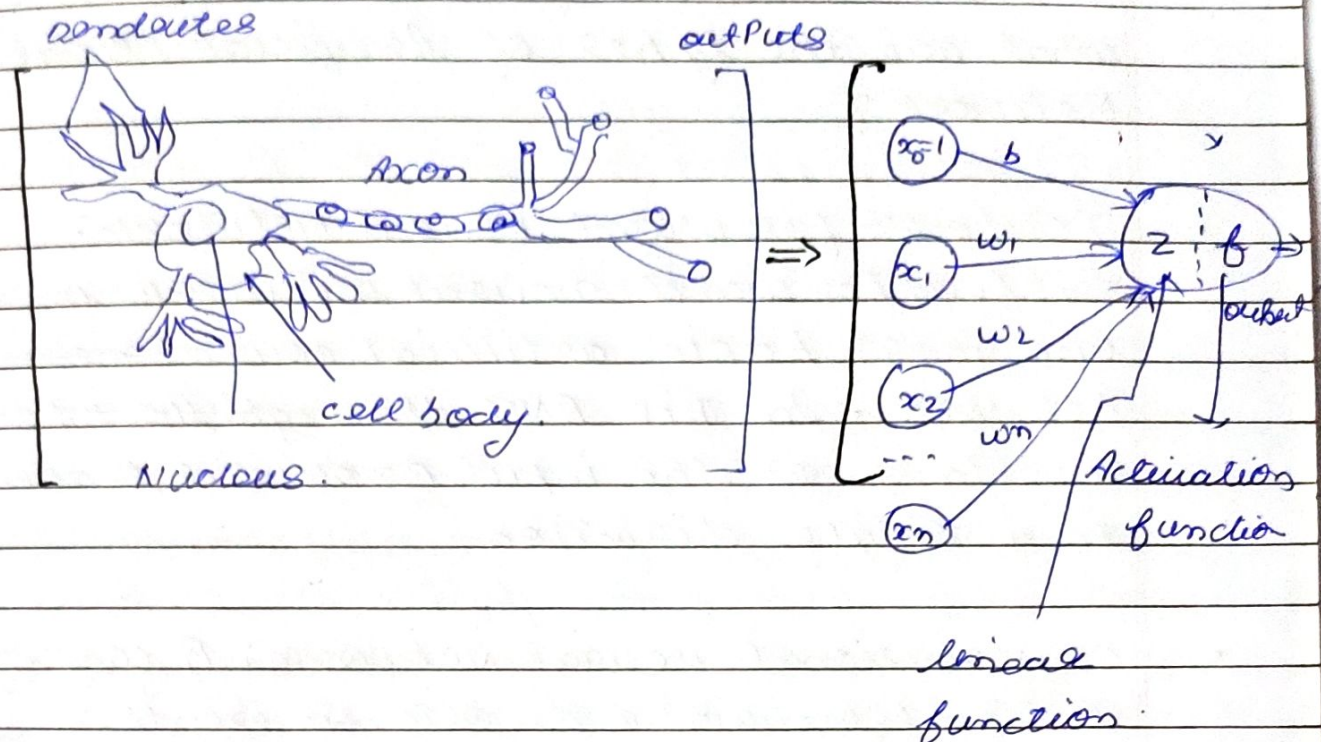
Solut. $\{(10, 0.08), (20, 0.04), (25, 0.63), (30, 0.18), (40, 0.6)\}.$

ARTIFICIAL NEURAL NETWORKS.

- Artificial Neural network contain artificial neurons which are called units.
- These units are arranged in a series of layers that together constitute the whole Artificial Neural network in a system.
- A layers can have only a dozen units or millions of units as this depends on how the complex natural network has an input layer, an output will be required to learn the hidden patterns in the dataset.
- In majority of neural networks, units are interconnected from one layer to another.
- Each of these connections has weights that determine the influence of one unit on another unit.
- As the data transfers from one unit to another, the neural network moves and



more about the data which eventually results in an output from the output layer.



How do Artificial Neural Networks learn?

- Ans.
- Artificial neural networks are trained using a training set.
 - For Example Suppose you want to teach an ANN to recognize a cat.
 - Then it is shown thousands of different images of cats so that the network can learn to identify a cat.
 - Once the neural network has been trained enough using images of cats, then you need to check if it can identify cat images correctly.



Date ___/___/___

- This is done by making the ANN classify the images it is provided by deciding whether they are cat images or not.

What are the types of Artificial Neural Network?

- ~~Feed~~ feedforward Neural Network: The feedforward neural network is one most basic artificial neural network. In this ANN through the input the data or the input provided travel in a single direction.
- Convolutional Neural Network: A convolutional neural network, where the connection has some similarities to the feed-forward neural network. where the connections, where the connections between units have weight that determines the influence of one unit on another unit.
- ~~Radial~~ modular Neural network: Radial basis functions are those functions that consider the distance of a point concerning the center.
- Recurrent Neural network: The Recurrent Neural Network saves the output of a layer and feed this output back to the input to better predict the outcome of the layer.



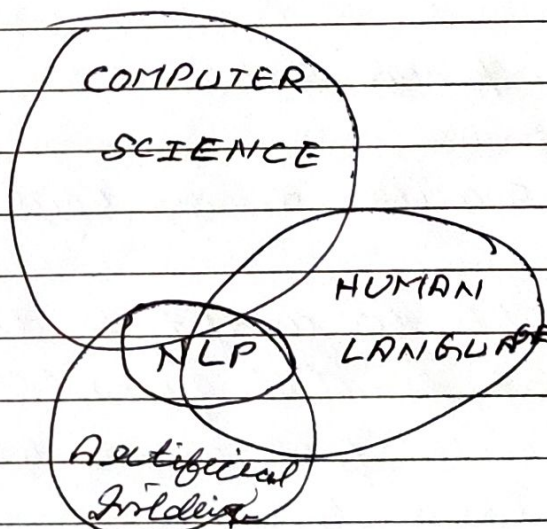
Date ___/___/___

Applications of Artificial neural network.

1. Social Media
2. Marketing and Sales
3. Health care
4. Personal Assistants

(NLP) Natural language Processing

NLP stands for Natural language Processing which is the part of computer science. Human language and Artificial Intelligence. It is a technology that uses machines to understand, analyse, manipulate and interpret human's language. It helps developers to organize knowledge for performing tasks such as translation, automatic summarization, Named Entity Recognition (NER), Speech recognition, Relationship Extraction, and topic Segmentation.





Date ___/___/___

Advantages of NLP.

- NLP helps users to ask questions about any subjects and get a direct response within seconds.
- NLP offers exact answers to the question means it does not offer unnecessary and unwanted information.
- NLP helps computer to communicate with human in their language.
- It is very time efficient.
- Most of the company use NLP to improve the efficiency of documentation processes accuracy of documentation and identify the information from large database.

Disadvantages of NLP.

- NLP may not show context.
- NLP is unpredictable.
- NLP may require more keywords.
- NLP is unable to adapt to the new domain and it has limited functions that's why NLP is built for a single and specific task only.



Components of NLP.

1. Natural Language Understanding (NLU).
Natural language understanding (NLU) helps to machine to understand and analyse human language by extracting the metadata from content such as concepts, entities, keywords, emotion, relations, and semantic roles.

NLU mainly used in Business applications to understand the customer's problem in both spoken and written language.

NLU involves the following tasks :-

- It is used to map the given input into useful representation.
 - It is used to analyze different aspects of the language.
2. Natural Language Generation (NLG) $\Rightarrow$ Natural Language Generation (NLG) acts as a translator that converts the computerized data into natural language representation. It mainly involves Text Planning, Sentence planning and text Realization.



NLU

- NLU is the process of reading and interpreting language.
- It produces non-linguistic outputs from natural language input.

NLG

- NLG is the process of writing or generating language.
- It produces constraining natural language outputs from non-linguistic inputs.

Supervised and unsupervised learning.

~~Supervised learning~~ $\Rightarrow$ Supervised learning, as the name indicates, has the presence of a supervisor as a teacher.

- Basically supervised learning is when we teach or train the machine using data that is well-labelled.
- which means some data is already tagged with the correct answer.
- After that the machine is provided with a new set of examples (data) so that the supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data.



For instance: Suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machines with all the different fruits one by one like this.

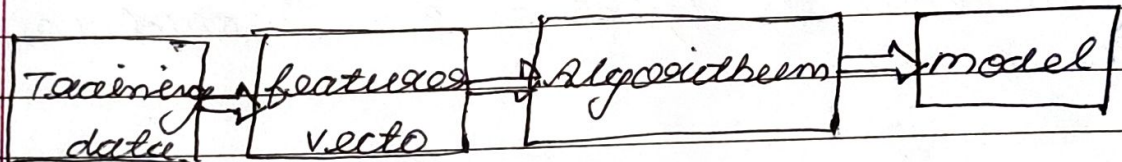
Advantages.

- Supervised learning allows collecting data and produces data output from previous experiences.
- ~~Helps~~ Helps to optimize performance ~~capability~~ with the help of experience.
- Supervised machine learning helps to solve various types of real-world computation problem.
- It performs classification and regression tasks.
- It allows estimating or mapping and ~~regression~~ The result to a new sample.
- We have complete control over choosing the number of classes we want in the training data.



Disadvantages.

- classifying big data can be challenging.
- training for supervised learning needs a lot of computation time. So, it requires a lot of time.
- supervised learning cannot handle all complex task in machine learning.
- computation time is vast in supervised learning.
- It requires a labelled data set.





unsupervised learning.

- unsupervised learning is the training of a machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance.
- Here the task of the machine is to group unsorted information according to similarities, patterns and differences without any prior learning of data.
- unsupervised learning is classified into two categories of algorithm.
- clustering $\Rightarrow$ A clustering problem is where you want to discover the inherent grouping in the data, such as ~~sort~~ grouping customers by their purchasing behaviour.
- Association $\Rightarrow$ An association rule learning problem is where you want to discover rules that describe large portions of your data. Such as people that buy X also tend to buy Y.



Supervised machine learning

- Algorithm are trained using labeled data.

- Simplex method

- High accurate

- NO. of classes is known

- uses offline analysis

- desired output is given

- use training data to infer model

- It is not possible to learn larger and more complex models than with supervised learning

- we can test our model

unsupervised machine learning

- Algorithm are used against data is not labeled

- computationally complex

- Less accurate

- NO of classes is not known

- use-real time analysis of data.

- desired output is not given

- NO training is used.

- It is possible to learn larger and even more complex model with unsupervised learning

- we cannot test our model



Advantages of unsupervised learning

- It does not require training data to be labeled.
- dimensionality reduction can be easily accomplished using unsupervised learning.
- Capable of finding previously unknown patterns in data.
- **Flexibility:** unsupervised learning is flexible in that it can be applied to a wide variety of problems, including clustering, anomaly detection, and association rule mining.
- **Exploration:** unsupervised learning allows for the exploration of data and the discovery of novel and potentially useful patterns that may not be apparent from the outset.
- **Low Cost:** ~~un~~ unsupervised learning is often less expensive than supervised learning because it doesn't require labelled data, which can be time-consuming and costly to obtain.



Disadvantages of unsupervised Learning.

- Difficult to measure accuracy or effectiveness due to lack of pre-defined answer during training.
- The results often have lesser accuracy.
- The user needs to spend time interpreting and label the classes which follows that classification.
- Lack of guidance: unsupervised learning lacks the guidance and feedback provided by labeled data, which can make it difficult to know whether the discovered patterns are relevant or useful.
- Scalability: unsupervised learning can be computationally expensive, particularly for large datasets or complex algorithms, which can limit its scalability.



Genetic Algorithm.

- Genetic Algorithm (GAs) are adaptive and heuristic search algorithms that belong to the larger part of evolutionary algorithms. are based on the ideas of natural selection and genetics.
- They are commonly used to generate high-quality solutions for optimization problem and search problems.
- Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation.

The whole algorithm can be summarized as -

1. Randomly initialize Population P
2. Determine fitness of Population
3. until convergence repeat:
 - a) Select parents from Population.
 - b) crossover and generate new Population.
 - c) Perform mutation on new Population.
 - d) calculate fitness of new Population.



Date ___/___/___

Example Problem and Solution using Genetic Algorithm.

Given a target string, the goal is to produce string from a random string of the same length. In the following implementation, following analogies are made.

- Characters A-Z, a-z, 0-9 and other special symbols are considered as genes.
- A string generated by those characters is considered as chromosome/solution individual.

Fitness score is the number of characters which differ from characters in target string at a particular index. So individual having lower fitness value is given more preference.

NOTE: Every time algorithm start with random strings, so output may differ



Constraint Satisfaction Problem (CSP) in Artificial Intelligence.

Finding a solution that meets a set of constraints is the goal of constraint satisfaction problem (CSPs), a type of AI issue. Finding values of a group of variables that fulfill a set of restrictions of rules in the aim of constraint satisfaction problem.

There are mainly three basic components in the constraint satisfaction problem.

variables: The things that need to be determined. and variables. variables in a CSP are the objects that must have values assigned to them in order to satisfy particular set of constraint.

Domains: The range of potential value that a variable can have is represented by domains. Depending on the issue, a domain may be finite or limitless.

constraints: The guideline that control how variables relate to one another are known as constraints.



Date ___/___/___

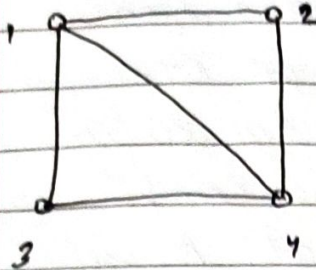
Constraint Satisfaction Problem (CSP)

Representation:

- The finite set of variables $V_1, V_2, V_3, \dots, V_m$.
- Non-empty domain for every single variables $D_1, D_2, D_3, \dots, D_m$.
- The finite set of constraints $C_1, C_2, \dots, C_m$.
- where each constraint C_i restricts the possible value for variables.
 - e.g. $V_1 \neq V_2$
- Each constraint C_i is a pair $\langle \text{Scope}, \text{relation} \rangle$
 - Example: $\langle (V_1, V_2), V_1 \text{ not equal to } V_2 \rangle$
- Scope = Set of variables that participate in constraint.
- Relation = list of valid variables value combination.
 - There might be a clear list of permitted combinations. Perhaps a relation that is abstract and that allow for membership testing and testing.



Constrained Satisfaction Problem (CSP)



$$V = \{1, 2, 3, 4\}$$

$$D = \{\text{Red, Green, Blue}\}$$

$$C = \{1 \neq 2, 1 \neq 3, 1 \neq 4, 2 \neq 4, 3 \neq 4\}$$

	1	2	3	4
Initial Dom	R, G, B	R, G, B	R, G, B	R, G, B
1 = R	R	G, B	G, B	G, B
2 = B, G	R	G	G, B	G, B
3 = G	R	G	G	B
4 = B	R	G	G	B

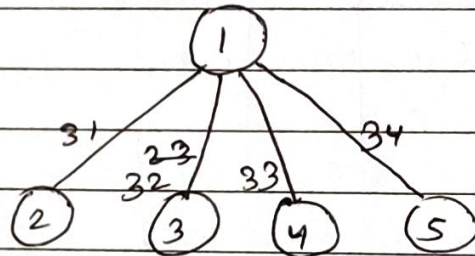


What is Branch and Bound Algorithm?

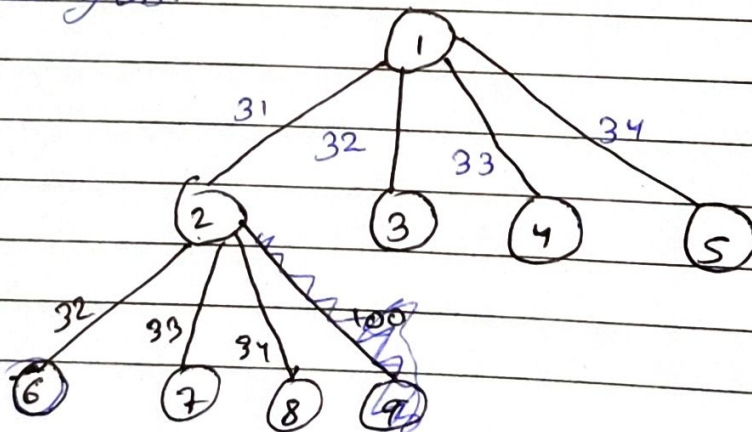
Branch and bound is an algorithm design Paradigm which is generally used for solving combinatorial optimization problems. These problems are typically exponential in terms of time complexity and may require exploring all possibilities in worst case. The branch and bound Algorithm techniques solve these problems relatively quickly.

First, we will see the subset method where we will see the variables size.

First method.

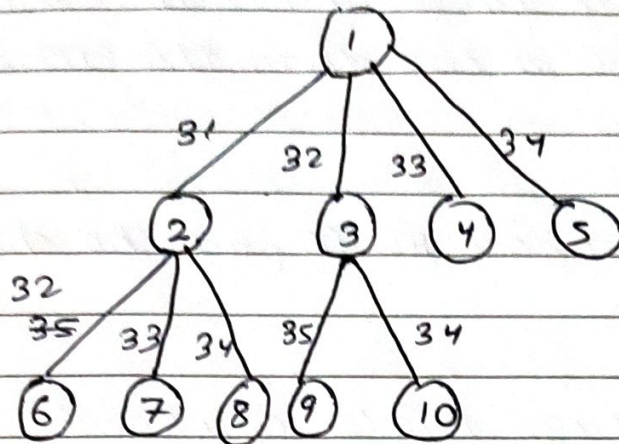


In this case, we first consider the first job, then second job, then third job and finally we considered the last job.

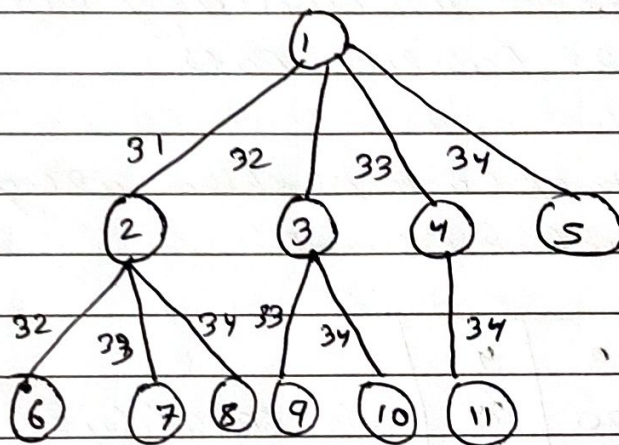




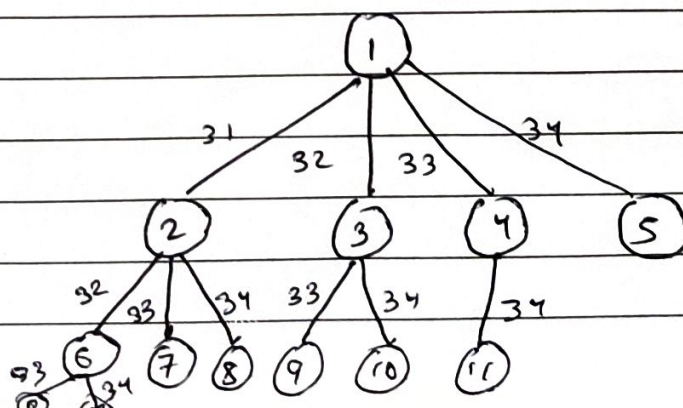
Now we will consider the node 3. In this case, we are doing job 2. So we can consider either j_3 and j_4 . Here we have discarded the job 1.



Now we will expand the node 4. Since here we are doing j_3 so we will consider only job j_4 .



Now we will expand node 6, and here we will consider the job j_3 and j_4 .





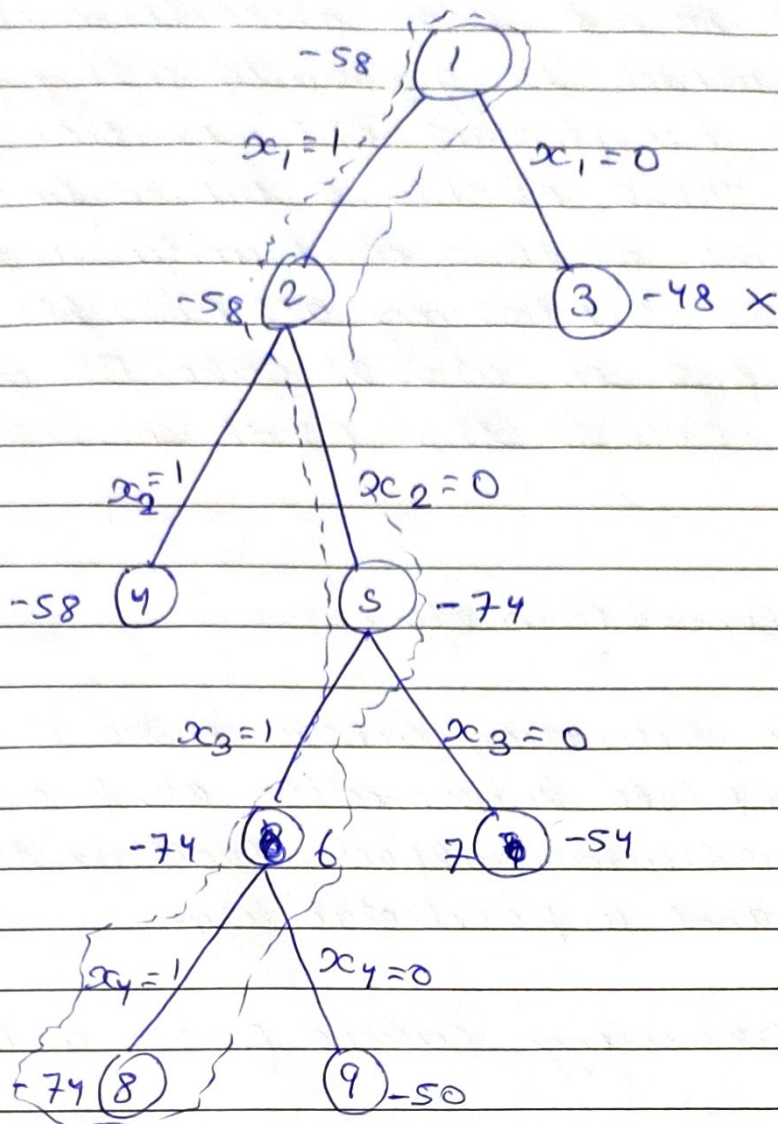
Algorithm ..

- Branching is process of generating subproblems
- Bounding refers to ignoring partial solution that cannot be better than the current best solution.
- It is a search procedure to find the optimal solution.
- It eliminates those parts of a search space which does not contain better solution (Pruning)
- In this method we basically extend the cheapest partial path.

0/1 Knapsack problem using
Branch & Bound.

i	1	2	3	4
P	30	28	20	24
w	5	7	4	2

capacity = 12.



2	28
1	30
58	

2	28
3	20
1	30

3	20
2	28
48	

4	24
3	20
1	30
-74	

4	24	4	24
3	20	1	30
1	30		
74		54	



Branch and bound is an algorithm design Paradigm which is generally used for solving combinatorial optimization problems. These problems typically exponential in terms of time complexity and may require exploring all possible permutations in worst case. Branch and bound solved the problem relatively quickly.

Uncertainty.

It refers to situation where there is a lack of complete information or knowledge about a particular aspect, leading to ambiguity and unpredictability.

Example: Drawing cards from a deck.

- Calculate the probability of drawing a red card (heart or diamond) from the deck ? ? ? ? ?
- Now let's introduce a condition: You want to calculate the probability of drawing a diamond given that you've already drawn a red card.

Let's represent the events:

A: Drawing a diamond (Event of Interest)

B: Drawing a red card (Condition)



calculate $P(A/B)$, which means "The Probability of drawing a diamond given that we've drawn a red card".

Reasoning under uncertainty.

- Various types of reasoning strategies used to handle uncertainty.
- 1. Probabilistic model.
- 2. Bayesian Networks.
- 3. Monte Carlo methods.
- 4. Decision Theory.
- 5. Fuzzy logic.
- 6. Qualitative Reasoning.