312823205107

M. Sudharsan IT-'B'

UNIT-1

PROBLEM SOLVING

1) A robot is tasked with finding a path from a starting point to destination in an unknown terrain represented the graph

a) Implement and demonstrate the working of UCS to find the optimal path

b) compare the UCS algorithm with BFS and DFS for the same problem.

ANS:-

Robot Navigation is a generalization of the route finding problem. A robot can move in a continous space with an infinite set of possibles action and states. For a circular robot moving on a float surface space is essentially 2D. when the robot has arms and legs or wheel then it must be controled in many dimensions.

1) states:

Locations, positions.

2) Initial state:
start (depends on task)

3) Function :

Movements

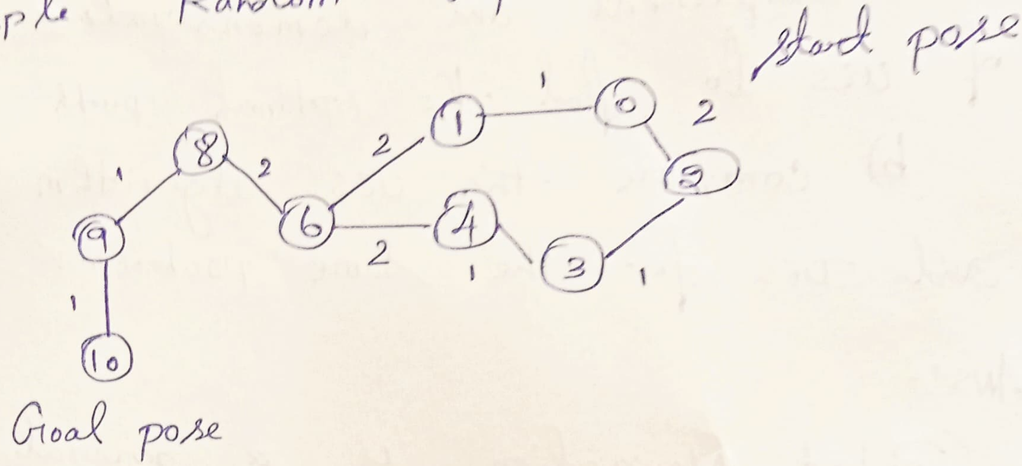
4) Goals

Tasks

5) path :

Distance, Energy.

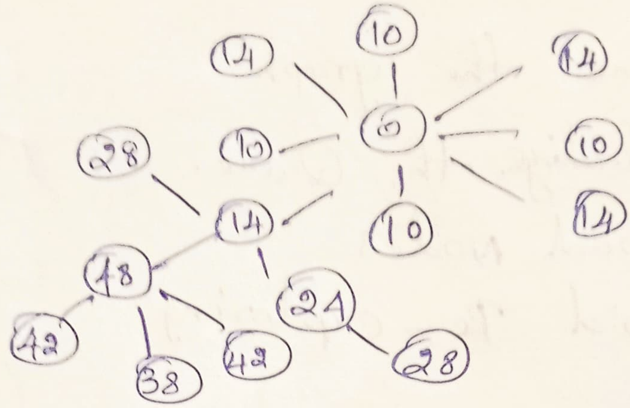
Simple Random representation :



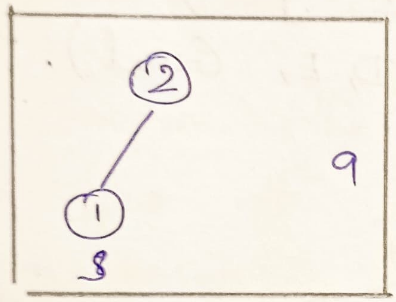

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 54 | 44 | 34 | 24 | 14 | 10 | 14 | 24 |
| 50 | 40 | 30 | 20 | 10 | 9 | 10 | 20 |
| 54 | 44 | 34 | 24 | 14 | 10 | 14 | 24 |
| 58 | 48 | 38 | 42 | 24 | 20 | 24 | 28 |
| 62 | 52 | 42 | 38 | 34 | 30 | 34 | 32 |
| 60 | 56 | 52 | 48 | 44 | 40 | 44 | 48 |

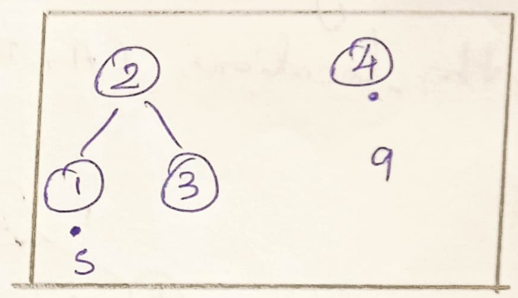
→ Optimal path

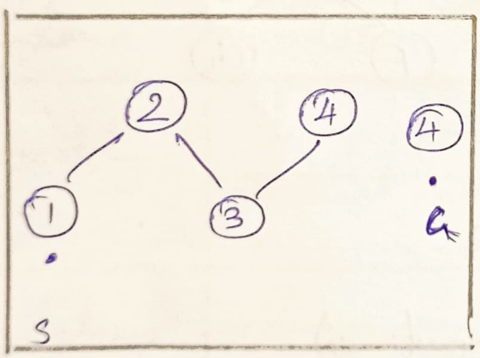
A* Search

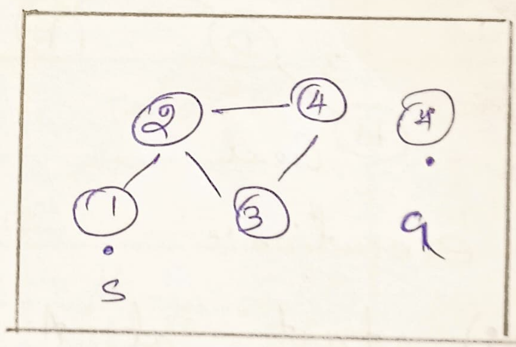



∴ No need to search fall path



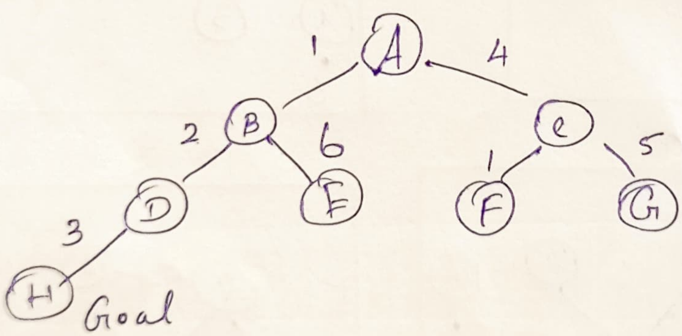

a) uniform cost search (UCS) is an uninformed search algorithm that finds the least cost path from start node to goal node & it is similar to Dijkstra's algorithm and ensures the optimal path in a weighted grams:

Steps:-

1. Define the graph
2. Initialize the Queue.
3. Expand Nodes
4. Avoid Re-exploring.

Example:-

Imagine a robot navigating a warehouse with locations (A, B, C, D, E, Goal). The costs are:




execution:-

i) Start at A (cost 0).

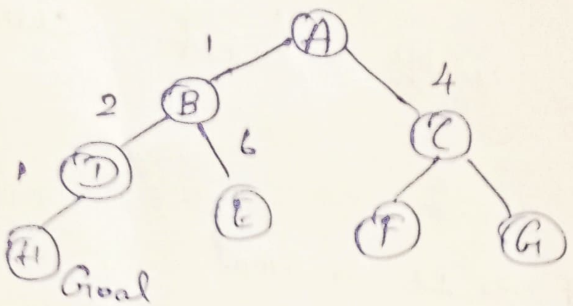
Expand to B (cost 1)

Expand to C (cost 4)

ii) B is taken as Parent due to low cost (1)

Expand B(1) → D (3), E (6)

D is taken due to its low cost.




step (iii) Expand D:

H is taken

∴ Goal node is achieved.

b) Compare the UCS algorithm with BFS and DFS for the same problem:

comparison:

| Criteria | UCS | BFS | DFS |
|---|---|---|---|
| optimality | Always finds the least co path | finds the shortest path | Does not guarantee optimality |
| completeness | Always completes if a solution exists | Always completes if a solution exists | May get stuck in infinite loops |
| time complexity | $o(b^d)$ | $o(b^d)$ | $o(b^d)$ |
| Space complexity | $o(b^d)$ | $o(b^d)$ | $ob^d)$ |

1. UCS :-
expands node with least cumulative cost.

Path :- A → B → D → Goal

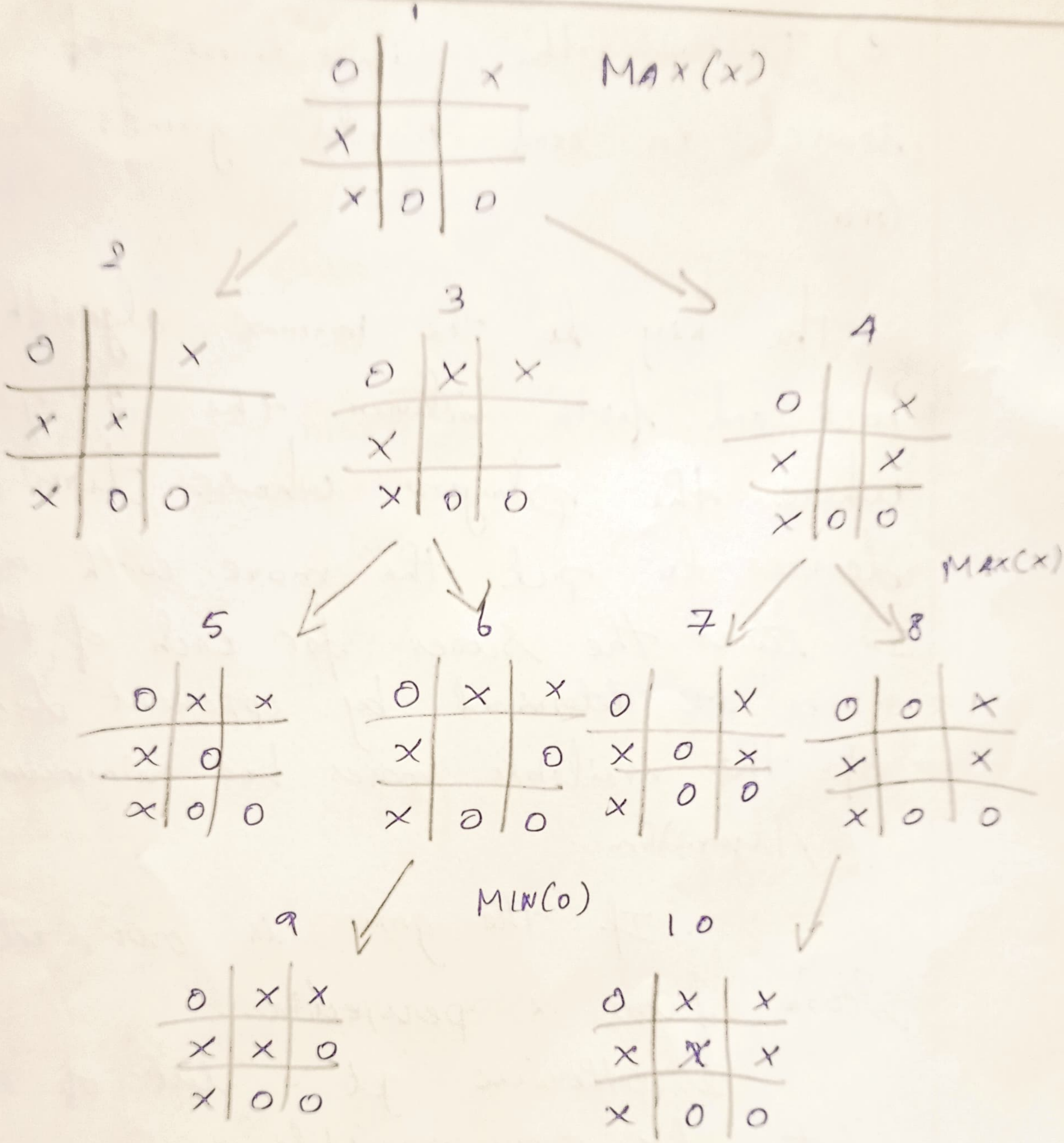
cost = 6

2. BFS :-
Expands all at current level

Path : A → B → D → Goal
All costs are so all but not optimal

3. DFS :-
Explores as deep as possible.
Possible path : A → C → B → Goal
cost = 7

2) Develop a two player game and implement the minmax algorithm for game.
a) extend the implementation with Alpha Beta pruning to improve efficiency
b) Compare the performance of minmax with and without Alpha beta pruning




The implementation of Alpha - Beta pruning with the min max algorithm for tic Tack. It focus on Creating a computer player that uses Alpha beta pruning is a search algorithm.

($\alpha$ - $\beta$) pruning is a search algorithm commonly used in game theory to improve decision-making algorithm.

Result:

c) Discuss the limitations of adversarial search in real world games like chess or Go.

The key to the minmax algorithm is a back and forth between the 2 players, where the player whose turn it is desires to pick the move with max score. In turns the scores for each of the available moves are determined by opponent decides which of the available moves has minimum score.

Algorithm:-

1. If the game is over, return the score from x perspective

2. otherwise get a list of new game states for every possible moves.

3. Create a score list.

4. For each states add minmax result of the state

5. If it x's turn, return the max score

6. If its O's turns return the min score

Let's show why algorithmically. The instant winning move will be picked.

b) MinMax Vs $\alpha - \beta$ pruning

| Features | Without purning | with pruning |
| --- | --- | --- |
| Nodes Explored | more | Fever |
| Time complexity | $O(b^d)$ | $O(b^{d/2})$ Best case |
| Space complexity | $O(b^d)$ | $O(bd)$ Same as min max |
| optimality | Always optimal | Always optimal |
| Performance | slower | faster. |

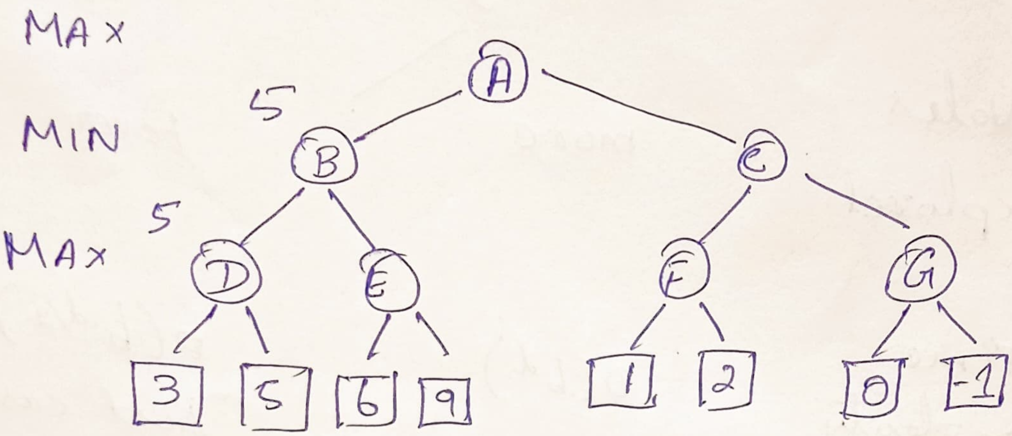
c) Limitations:-

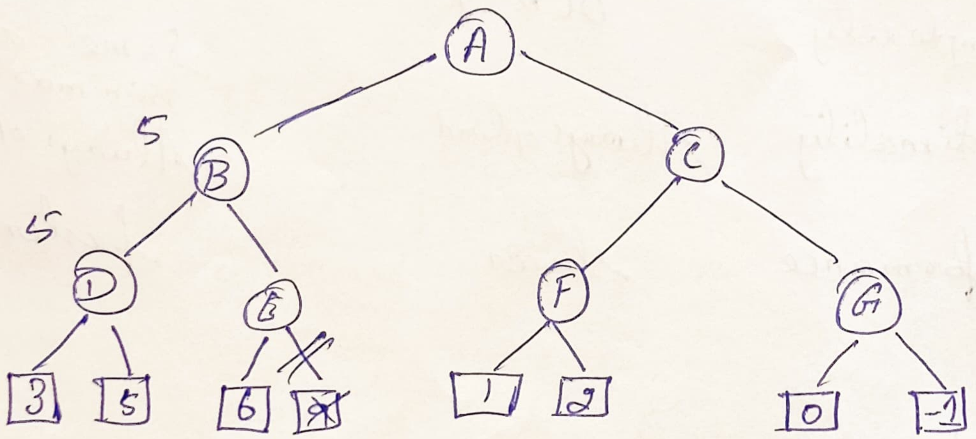
1. High computational complexity: Games like chess or Go have enormous search spaces making brute-force MinMax.

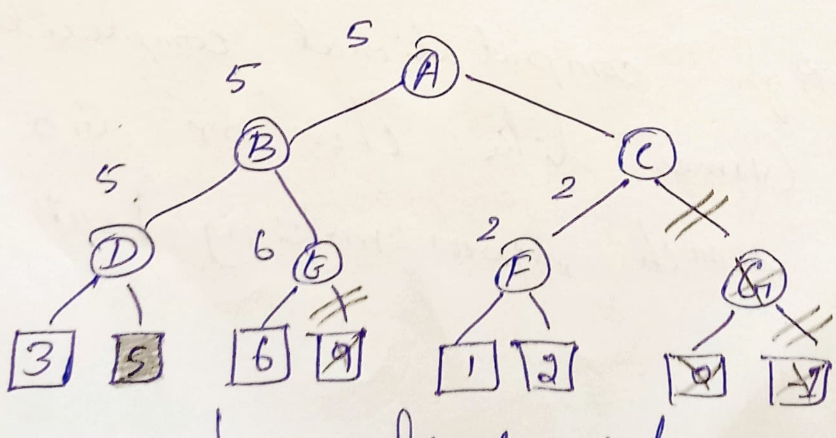
1) -1 if the player 'o' that seeks min wins

2) 0 if its a tie

3) 1 if player 'x' wins ⇒ max wins ;




The optimal value is : 5

2. Depth limitation:

Due to time constraints the search depth is limited leading a sub optimal decisions

3. Heuristic Dependency

Strong AI relies heuristics which may ~~or~~ not have accurate.

4. Handling uncertainity

5. Memory constraints

6. Real - Time Decision making

7. Human Intution Gap:

AI lacks creativity and long-term planning compared to human intuition.

These are the some of the limitations of Adversial Search in Games like chess or Go

May be in future the limitations will be resolved.

# UNIT-2

## PROBABILISTIC REASONING

1) Design a Spam email classifie using the Naive Bayes Algorithm.

a) Explain the assumptions behind the Naive Bayes model

b) Implement the algorithm to classify emails into "Spam" or "Not Spam" based on word frequencies

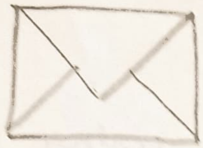
Naive Bayes classifies are supervised machine learning algorithms used for classification tasks, based on Bayes theorem to find probabilities.

Naive - assume the presence of one features does not affect other.

Bayes - refers to basis in Bayes theorem.

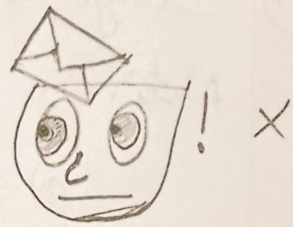
| S.NO | outlook | temperature | Humidity | play Golf |
|---|---|---|---|---|
| 0 | Rainy | HOT | High | NO |
| 1 | Rainy | HOT | High | NO |
| 2 | Overcast | Mild | ~~High~~ | Yes |
| 3 | Sunny | cool | Normal | Yes |
| 4 | Sunny | cool | Normal | Yes |
| 5 | Sunny | Hot | High | NO |

Email



Machine Learning Model

Calculation helps to understand:



$$S[T] = \frac{C_{spam}(T)}{C_{spam}(T) + C_{Ham}(T)}$$

Where $C_{spam}(T)$ and $C_{Ham}(T)$ are the no. of spam or ham messages containing token. T, respectively. To calculate the possibility for a message M with tokens $\{T_1, \ldots \ldots T_n\}$, one needs to combine the individual tokens spamminess to evaluate the overall message spamminess

2 stages:-

1. Training
2. Testing

a) Assumptions:-

1. conditional Independence Assumption:-

Features are assumed to be independent of each other given the class label. This simplifies computation but may not always hold in real-world data.

2. Features Relevance:

Each features contributes independently

to the probability of a given class

3. Equal importance of features:-

All features contributes equally

4. Data follows a distribution.

i) Gaussion Naive Bayes

ii) Multinomial Naive

iii) Bernoulli Naive

b) Implementation:-

1. preprocess the Emails:

Tokenize and clean Text

2. Calculate word Frequencies:

compute probabilities of words given Spam and not spam.

3. Apply Bayes Theorem:-

use probabilities to classify new emails.

List of stop words:-

["i", 'me', 'my', 'him', 'his', 'himself', 'it', 'its', 'itself', 'they', 'them', 'myself', 'did', 'doing', 'to', 'from', 'further', 'most', 'only', 'will', 'can', 'just', 'done' .... etc...]

4 parameters:-

- Accuracy: (TP+TN)/(TP+FP+FN+TN)
- precision: TP/(TP+FP)
- Sensitivity: TP/(TP+FN)
- Specify: TN/(TN+FP)

Gmail, yahoo, outlook:

Gmail:

Goggle data centers use thousands of rules to filter spam emails. They provide the weightage to different parameters based on that they filter the emails. It is an "state of an art technique".

yahoo:-

It is the world's first free webmail service provide. with more than 320M active users. They have their own filtering techiques to emails. Basic methods are:

1) URL filtering
2) email content
3) user spam complaints

Outlook:

Microsoft owned mailing platform microsoft renamed hotmail and windows live mail to outlook. It has 400M activer users.

It uses

1) Safe sender list
2) Safe Recipients list
3) Blocked senders
4) Blocked Top level Domains
5) Blocked Encodings.

2) using Crypt Arithmetic expression solve

BASE
BALL
GAMES

Conditions :-

1) Each character should have unique value
2) Digits should be (0-9)
3) starting character cannot be 0
4) only one solution should be there
5) If you are adding two same character then our output should be Even
6) Incase of addition of 2 numbers it there is a carry to next step then the carry be only one.

Solving the expression:

| | | | | | |
|---|---|---|---|---|---|
| | 9 | | | | |
| | B | A | S | E | |
| (+) | B | A | L | L | |
| (1) | G | A | M | E | S |

B - 5
A - 0
S
E
L
G - 1
M

i) G = 1

B - 6
A - 2
S
E
L
G - 1
M

ii)

| | | |
|---|---|---|
| | 1 | |
| 5 | 0 | |
| 5 | 0 | B ≥ 5 α |
| 1 0 | ~~0~~ , | |

iii)

| | | |
|---|---|---|
| | 6 | 2 |
| | 6 | 2 |
| 1 | 2 | 4 |

3rd column:

$S + L = E$

(or)

$S + L = B + 10$

4th column:

$B + L = S$

$B + L = S + 10$

$\cancel{S} + L + L = \cancel{S} + 10$

$2L = 10$

$L = 5$

$$\begin{array}{r} B = 7 \\ + \quad L = 5 \\ \hline S = 12 \end{array}$$

$S = 2$

iv)

| | | | | |
|---|---|---|---|---|
| | 6 | 2 | | |
| | 6 | 2 | 5 | 5 |
| 1 | 2 | 4 | | |

3rd column:

$S + 5 = B$

$B + 5 = S$

$(8, 3)(9, 4)$ ×

$S = 8, B = 3$

V) 7 4 8 3

7 4 5 5

1 4 9 3 8

∴ $B=7$, $A=4$, $S=8$, $B=3$, $L=5$,

$a=1$, $M=9$.

∴ All conditions satisfied.

Advantages :-

1) Enhances logical thinking
2) strengthens Mathematical skills
3) Encourages Algorithmic thinking
4) Improves Attention to detail.
5) Fun and Engaging.
6) useful in exams.

Disadvantages :-

1) Time consuming
2) Limited practical use
3) High complexity
4) Frustration for Beginners