Name : Abhishek Srivastava.
Reg. No. : 19BCE10071
Subject : Artificial Intelligence.
Slot : C11
Date : 06/08/21
Class No : 0040

---

1) A Heuristic is a technique to solve problem faster than classical methods, or to find an approximate solution when classic method cannot. This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed.

A Heuristic (or heuristic function) takes a look at search algorithms. At each branching step, it evaluates the available information and makes a decision on which branch to follow.

We need Heuristic Algorithms to produce a solution, in reasonable amount of time that is good enough for the problem in question

This reduces exponential problems to polynomial time complexity.

Some of the Heuristic search techniques are :-

- Best - First Search
- A^* Search
- Hill Climbing

Best - First Search.

Best-First Search is an Instance of a general TREE-SEARCH or GRAPH-SEARCH algorithm.

A node is selected for expansion based on an Evaluation function, $f(n)$. This Evaluation function is constructed as a cost estimate, so the node with lowest evaluation will be expanded first

Most Best-first search algorithm include as a component of f a heuristic function, denoted $h(n)$.

$h(n)$ = Estimated cost of a cheapest path from the state at node n to the goal.

③

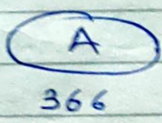
One of the real world applications of Best-first search can be finding path from one place to another.

Straight line Distance heuristic.

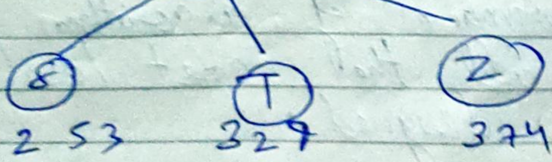
A	366	M	241
B	0	N	234
C	160	O	380
D	242	P	100
E	161	R	193
F	176	S	253
G	77	T	329
H	151	U	80
I	226	V	199
L	244	Z	374

(a) Initial State.

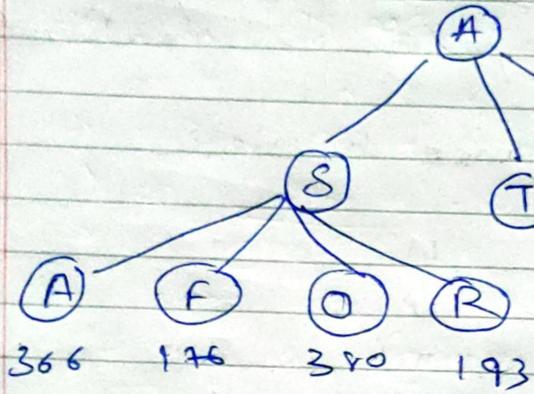
Here is Best-first search tree from B node.



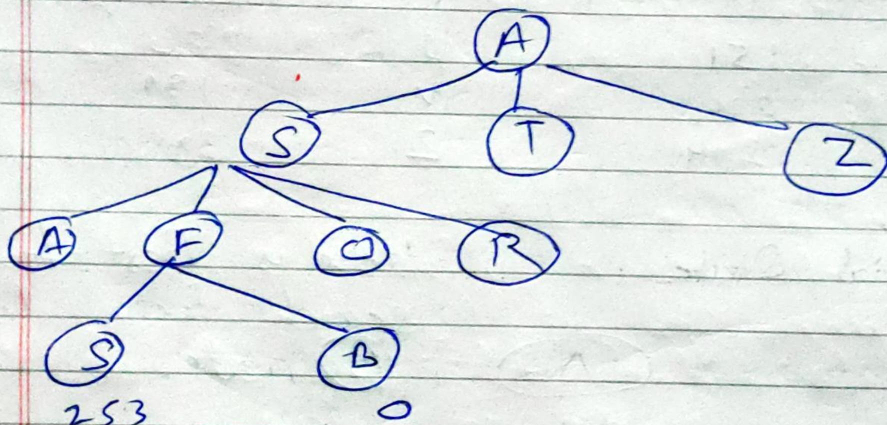
(b) After Expanding A



(c) After Expanding S



(d) After Expanding F



Worst Case time Complexity is $O(n * \log n)$ when n is number of node.

Performace of algorithm depends on how well the cost evalution function is designed.

(2) (b) $A = \{(1, 0.1), (2, 0.5), (3, 0.8), (4, 1.0), (5, 0.7), (6, 0.2)\}$.

Normalised fuzzy set:

$$\mu_{\bar{A}_{norm}}(x) = \frac{\mu_{\bar{A}}(x)}{\sup_x \mu_{\bar{A}}(x)}$$

The support of a fuzzy set (denoted Supp) is the crisp set of all $x \in X$ for which $\mu_{\bar{A}}(x) > 0$.

$$supp(\bar{A}) = \{1, 2, 3, 4, 5, 6\}$$

The core of a fuzzy set is the crisp set for which $\mu_{\bar{A}}(x) = 1$

$$core(\bar{A}) = \{4\}$$

The α-cuts of the fuzzy sets are:-

$A_{0.1} = \{1, 2, 3, 4, 5, 6\} = supp\,\bar{A}$

$A_{0.2} = \{2, 3, 4, 5, 6\}$

$A_{0.5} = \{2, 3, 4, 5\}$

$A_{0.7} = \{3, 4, 5\}$

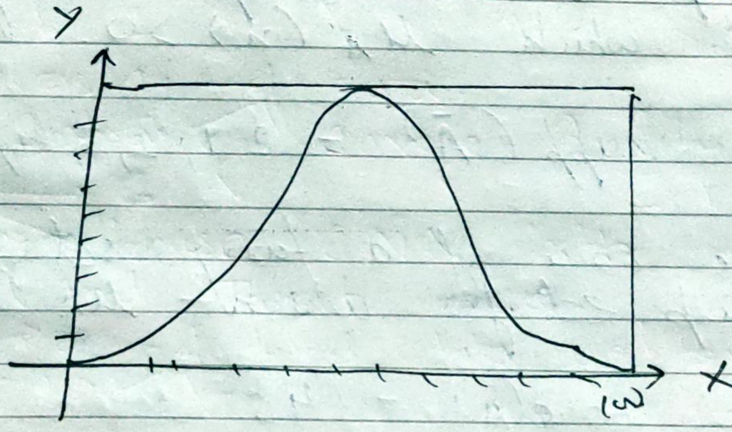
$A_{0.8} = \{3, 4\}$

$A_{1.0} = \{4\} = core\,\bar{A}$

We can define strong α-cuts as the crisp set A_α' for which $\mu_{\tilde{A}}(x) > \alpha$

(a) $$Y = \frac{1}{e^{c(|x-k|)}} = e^{-c(|x-k|)}$$

It is a Gaussian Membership function. ~~where~~



3) Machine Learning can solve numerous real-world problems. Some of them are :-

- Spam Detection
- Credit Card Fraud System
- Digit Recognition
- Speech Understanding
- Face Detection
- Product Recommendation
- Stack trading
- Shape detection.

Hand written Digit Recognition.

Handwriting Recognition (HWR), also known as Hand Written text Recognition (HTR), is the ability of a computer to recieve and interpret ~~intelligent~~ intelligible handwritten input from source such as paper documents, photographs, touch screens and other devices.

Important steps in Algorithm.

1) Preperation of Dataset.

→ Ask different people for writing digit 0 - 9 in a paper.
→ From each person 100 data can be collected.
→ If 100 people are involving, we have 10000 data.

2) Scanning and digitaizing data.

3) Preprocessing of data.
→ Improving Image Quality

4) Training phase CNN with input image and Target value.
→ 80% of total data used for training.

5) Testing phase
→ 20% data used for testing.

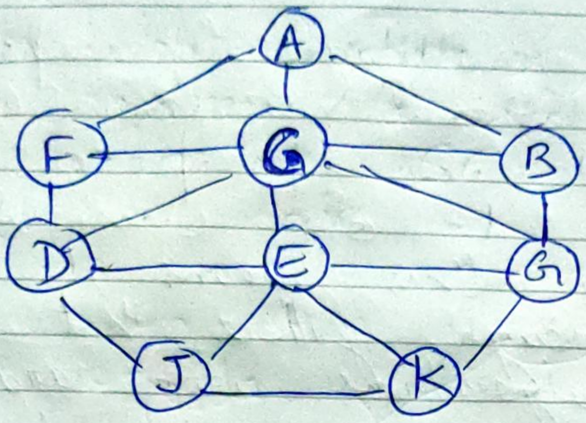
Algorithm Involved. → CNN

CNN is basically a model known to be Convolutional Neural Network

CNN uses multilayer perceptrons to do computational work. CNN's use little less pre-processing as compared to othe classical algorithms. This means the network learns through filters that is in traditional algorithms were hard-engineered. So, for image processing task CNN's are best-suited.

PTU

5)



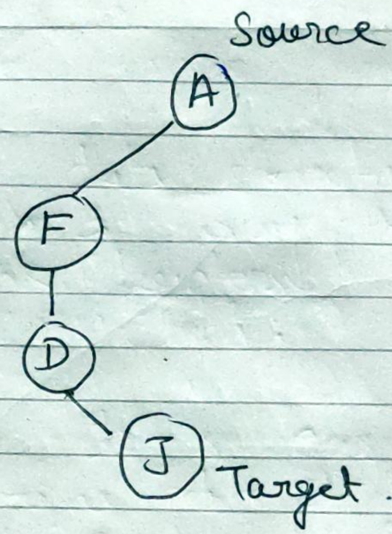
Find 3 different Solution. Searching J from source A.

Solution 1

A → F ✓, G, B

F → A, G, D ✓

D → F, G, E, J ✓



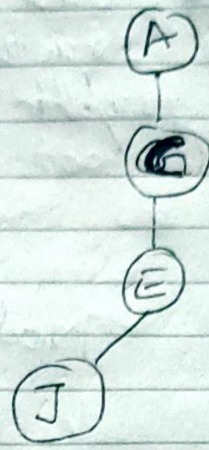
A → F → D → J

Solution 2

A → F, G ✓, B

C → F, A, B, G, E ✓, D

E → D, G, C, K, J ✓



A → C → E → J

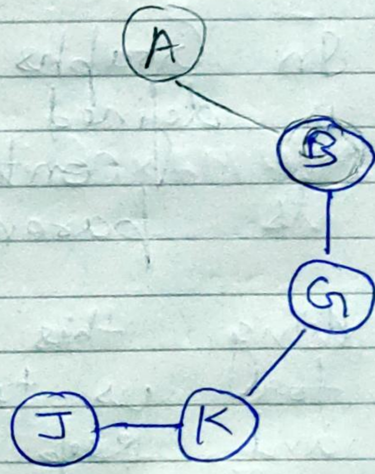
Solution 3

A → ~~A~~, G, B ✓

B → A, G ✓, C

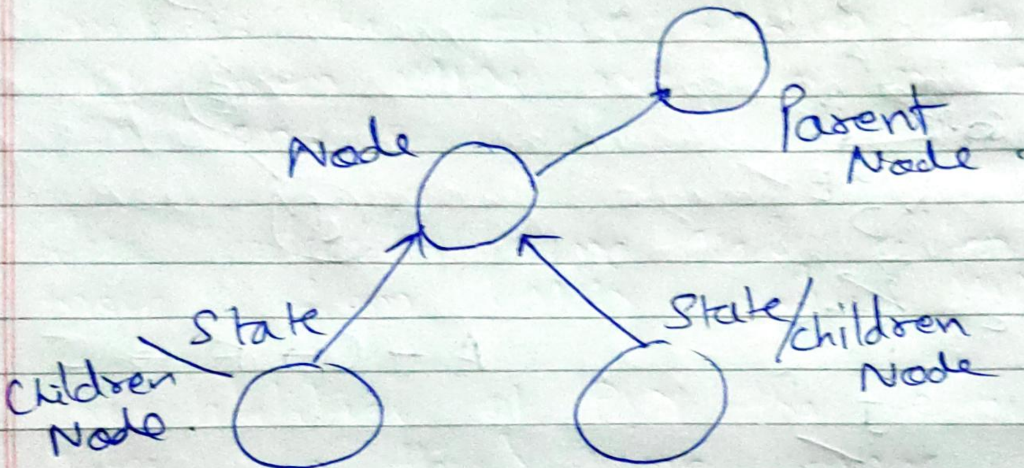
G → B, E, A, K ✓

K → G, E, J ✓



A → B → G → K → J

Nodes
~~Graph~~ data structure is most suitable data structure to store vertices of a graph in BFS algorithm



In Graphs, Node data structure is defined in such a way that it contains Link (pointers) to its parent and child.

Using this structure, it is easy for us to maintain Graph structure and keep track of each Node.

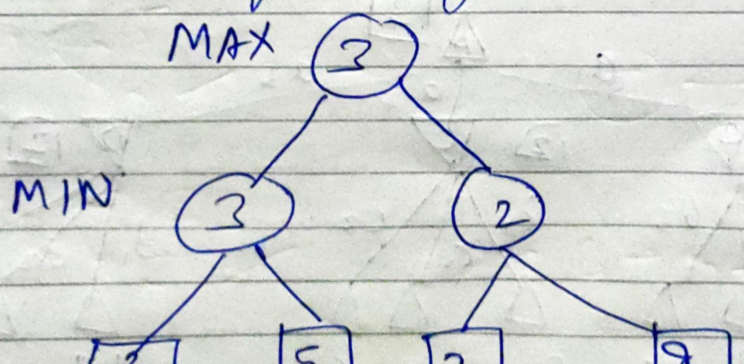
4) MINIMAX.

Minimax is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally.

It is widely used in games like tic-tac-toe, Backgammon, Mancala, chess, etc.

In MINIMAX, there are two players called maximiser and minimizer.

The minimizer tries to ~~to~~ get the lowest possible score and maximizer tries to get highest possible score.



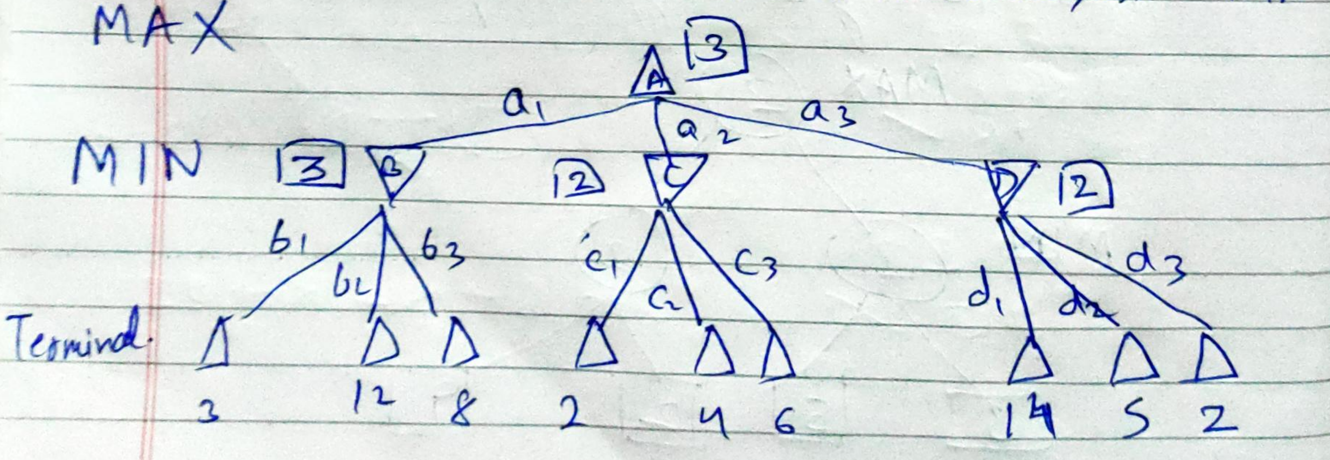
Two-ply Game Tree

The possible moves for MAX at the root node are labelled a_1, a_2 and a_3. The possible replies ~~for~~ to a_1 from MIN are b_1, b_2, b_3 and so on.

This particular game ends after one move from MAX and MIN.

The utilities of PLY the terminal states in this game range from 2 to 14.

$$\text{MINIMAX}(s) = \begin{cases} \text{UTILITY} & \text{Terminal} \\ \max \text{MINIMAX}(\text{Res}(s,a)) & \text{MAX} \\ \min \text{MINIMAX}(\text{Res}(s,a)) & \text{MIN} \end{cases}$$



Δ → Max Node
∇ → Min Node.

Explanation

B → 3 (which is min(3, 12, 8))
C → 2 (which is min(2, 4, 6))
D → 2 (which is min(14, 5, 2))

A → 3 (which is max(B, C, D)).

Optimal Strategy:

For MIN → to select minimum among all its children states

For MAX → to select maximum of all its children state.