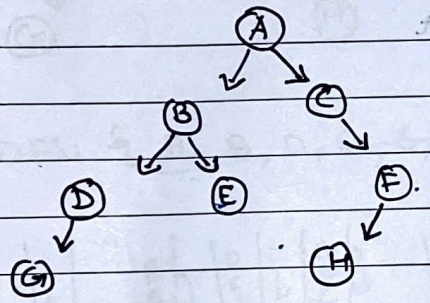
Page No.:
Date:
YOUVA

BFS

Algorithm :

1) Enter starting node on queue.
2) If queue is empty, return "Fail" and stop.
3) If first element on queue is goal node, then return "Success" and stop.
4) Else, remove and expand first element from queue and place children at end of the queue.
5) Goto step 2



b^d

b = Branching = 2

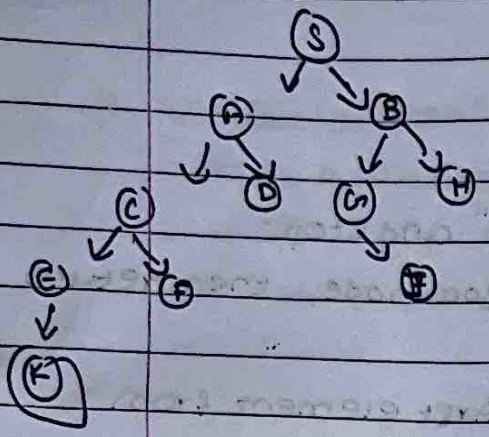
d = depth = 3 $\Rightarrow 2^3 \Rightarrow 8$

Time complexity O(8)

Path: A B C D E F G H

A	
BC	A
CDE	AB
DEF	ABC
EFG	ABCD
FG	ABCDE
GH	ABCDEF
H	ABCDEFG
	ABCDEFGH

Q.



Here goal node is K

Queue	Output
{ S }	-
{ A, B }	S
{ B, C, D }	SA
{ C, D, G, H }	SAB
{ D, G, H, E, F }	SABC
{ G, H, E, F }	SABCD
{ H, E, F, J }	SABCDG
{ E, F, J }	SABCDGH
{ F, J } K }	SABCDGHE
{ J, K }	SABCDGHEF
{ K }	SABCDGHEFJ
-	SABCDGHEFJK

The final path of traversal is:

SABCDGHEFJK

Thus time complexity is: b^d

$= 2^4$

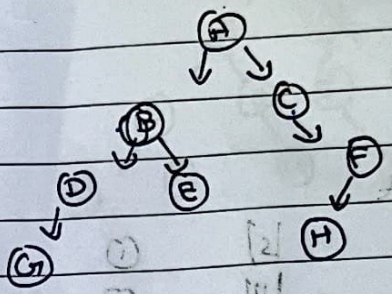
$= 16$

$= O(16)$

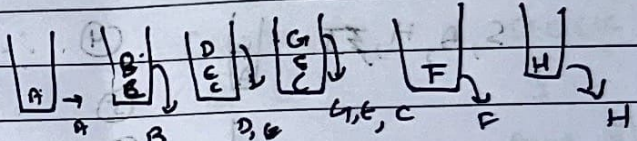
space complexity = b*d = 8,

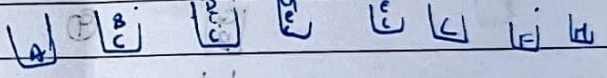
DFS Algorithm

1. Enter root node on stack.
2. Do until stack is not empty.
3. Remove node:
   - i. If node is goal node, STOP
   - ii. Else push all children node in stack.

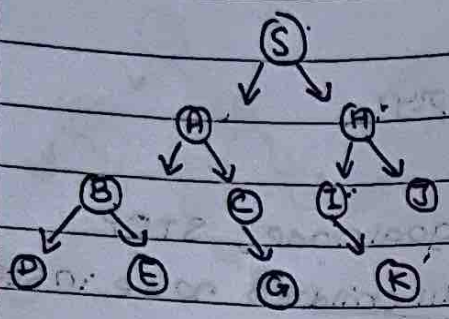


DFS: A, B, D, G, E, C, F, H





Q.



Here goal node is G

Stack	Output		
S	—	S	(1)
A, H	S	S H A	(2)
H	S, A	S H A	(3)
I	S, A, H, I	S H A I K	(4)
A, K		S H A I K D	(5)
D		S H A I K D	(6)
B, C		S H A I K D B C	(7)
B, G		S H A I K D B C G	(8)
B		S H A I K D B C G	(9)

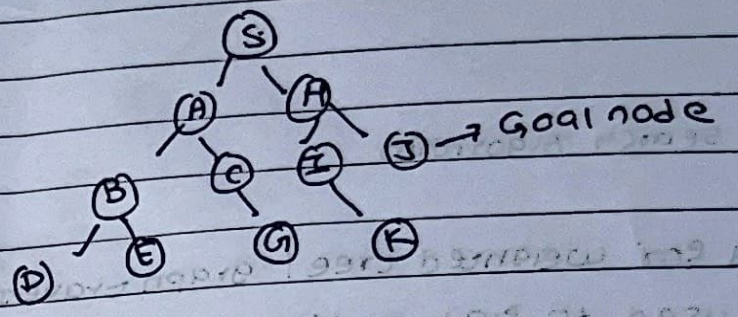
Path of traversal:

S H I I K A C G

Page No.:
Date:

Depth Limited Search Algorithm

- working is similar to DFS with a predetermined limit.
- Helps in solve problem of DFS (i.e. DFS goes to infinite path)



J is goal node

d=2

∴

Output	Stack
&	S
S	H A
S H J	J H A
~~S H J~~ S H J	

Path of traversal: ~~S H J~~ S H J

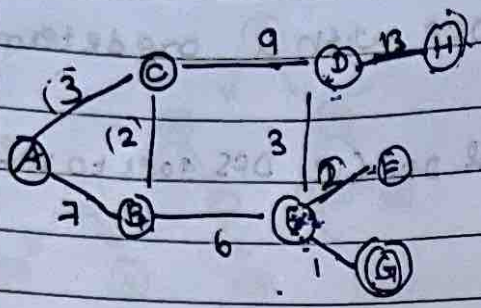
① S → H A → J J H A → J H A

S → H → J

Time complexity = $b^d = 2^2 = 4 = O(4)$

Space complexity = 2*2 = 4

Q.



A C B E G E F G D H

3 + 2 + 6 + 1 + 1 + 2 + 2 + 3 + 13

= 33 //

(A) Uniform cost search Algorithm

1) It is used for weighted tree / graph traversal
2) It's goal is used to find path to the goal node with lowest cumulative cost. (minimum)
3) Advantage: Optimal solution
   Disadvantage: Stuck in infinite solution
4) uses priority queue.

Solution :

A C B E G

Total cost: 3 + 2 + 6 + 1 = 12

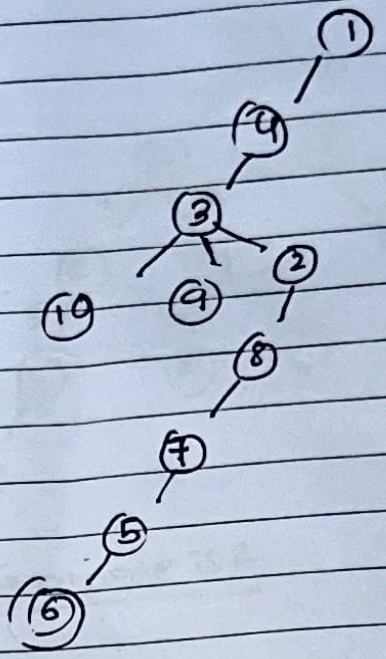
Now if goal node is A,

Path of Traversal: A C B E G E F E D H

= 33

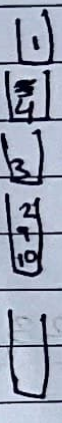
Q.

DFS



Ans

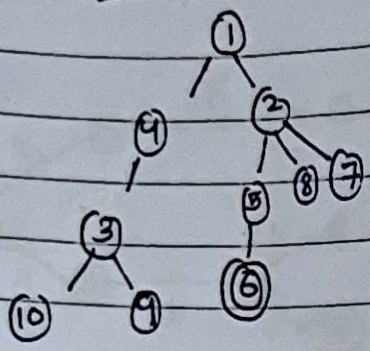
Stack	Output
1	-
4 3 2	1
3	14
10,9,2	143
10,9,8	1432
10,9,7	14328
10,9,5	143287
10,9,6	1432875
	14328756



Path of traversal : 14328756

Q

BFS:



Queue	output
1	-
42	1
23	14
3587	142
587109	1423
87109 6	14235
7109 6	142358
109 6	1 4 2 3 5 8 7
9 6	1 4 2 3 5 8 7 10

1,4,2,3,5,8,7,10,9,6

1 -

4,2 1

2,3 14

3587 142

587109 1423

87109 6 14235

7109 6 14235

109 6 1423587

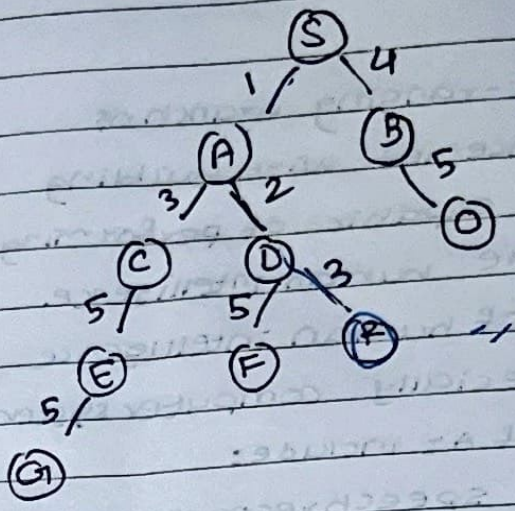
9 6 14235 87 10

9 6

Page No.:
Date:

YOUVA

UCS



Goal node is P

Path of traversal : S → A → D → P

Total cost = 1+2+3 = 6 //

Page No.:
Date:

AI - Unit 1 : Intro. to AI

⊙ Definition : • AI is a wide-ranging branch of computer science that is concerned with building smart machines that are capable of performing tasks that typically require human intelligence.
• AI is the simulation of human intelligence process by machines especially computer systems.
• Specific applications of AI include:
Expert Systems, ~~OR~~ NLP, speech recognition, Machine vision
• At its simplest form, AI is a field which combines computer science and robust datasets to enable problem solving.

⊙ Weak AI vs Strong AI

1] Weak AI:
- Aka Narrow AI or Artificial Narrow Intelligence (ANI) is AI trained & focused to perform specific tasks.
- Eg: Apple's Siri, Amazon's Alexa.

2] Strong AI :
- Made up of AGI & ASI.
- AGI : Artificial General Intelligence
  • Theoretical form of AI
  • Have intelligence of machines equalled to humans.
  • would have self-aware consciousness that has ability to solve problems, learn & plan for future.
- ASI : Artificial super intelligence would surpass intelligence & ability of human brain

Page No.:
Date:

⊛ Application of AI:

1) AI in Astronomy:
   ① Solve complex universe problems.
   ② Understanding universe & its working.

2) Healthcare:
   - Faster & better diagnosis than humans.

3) Gaming:
   - In games like chess where machines need to think of large no. of possible places.

4) Finance:
   - Implementation of automation, chatbot, algo trading etc in financial processes.

5) Data security:
   - Helps in making data safe & secure.
   - Used to determine software bug & cyber attacks.

6) Social Media:
   - Organize & manage massive amounts of data.
   - Used to identify latest trends, hashtags.

7) Travel & Transport:
   - Customer service, Travel arrangements

8) Entertainment: Recommendation

Page No.:
Date:

⊙ AI with NLP:

- NLP refers to AI method of communicating with an intelligent system using natural language
  eg: English.
- NLP is required when you want an intelligent sys. like robot to perform as per our instructions
- The input & output of an NLP system can be:
  i) Speech
  ii) written text.

TWO Components of NLP are:

1) NLU:
   - understanding ; involves following tasks:
   ① Mapping given input in NL into useful representations.
   ② Analysing diff-aspects of language.

2) NLG:
   - Produces meaningful sentence in form of NL from internal representation.
   - Involves:
     ① Text planning, ② Sentence planning, ③ Text realization

Page No.:
Date:

U2;

- In order to provide formal description of a problem it is necessary to do following things:-
  1) Define state space that contains all possible configs. of relevant objects.
  2) Describe initial states: states that describe possible situation from which problem solving starts.
  3) Specify one or more states that could be acceptable as solution to problem. These states are called goal states.
  4) Specify a set of rules that describe available actions.

- AI is the study of building agents that act rationally.
  Most of the time, these agents perform sort of search algorithm to achieve their tasks.
- Search problem consists of:
  i) State space: Set of all possible states where you can be
  ii) Start State: State from where search begins.
  iii) Goal Test: Functions that looks at current state returns whether it is/it goal state.
- Soln to search problem is sequence of action called plan that transforms from start state to goal state.
  - Plan is achieved through search algorithms.

Page No.:
Date:

⊙ Robotics

1. Assembly: When combined with advanced vision, robotics can help with real time course correction useful in aerospace.
2. packaging: Quicker, lower cost & more accurate packagings.
3. customer service: uses NLP, chatbots
4. open source Robotics: Small scale Agriculture as they are more smarter, accurate & profitable.

⊙ ML & DL

AI:
- Any techniques that helps computers to ~~create~~ mimic human intelligence using if-then rules, decision trees, ML, DL.

ML:
- A subset of AI that includes statistical techniques that enable machines to improve at tasks with experience.

DL:
- A subset of ML composed of algorithms that permit software to train itself to perform tasks like speech recognition by "exposing multilayered neural networks to vast amount of data".

Page No.:
Date:

⊙ Machine learning

⊖ Field of study that gives computer the ability to learn without being explicity programmed.

⊖ Automating & improving the learning process of computer based on their experience without actually being programmed.

⊖ Process starts with feeding good quality data & then training our machines using ML models via data & algos.

- Applications: Medicine, speech Recognition
- Example: Training of student during exam

⊙ DL:

- Subset of ML composed of algorithm that permits software to ~~improve~~ perform & train itself to perform tasks like speech recognition by exposing multilayered neural networks to vast amounts of data.

Perform test every time, the DL algorithm would tweak a little to improve the outcome.

- Example:

1) Virtual Assistants: uses DL to help understand the speech & language humans use.
2) Translation: DL algorithm can automatically translate in language.
3) Chatbot:
4) Image colourization
5) ~~Facial recognize~~ Medicine
6) OTT.

Page No.:
Date:

① Machine Learning vs Deep Learning

Machine Learning	Deep Learning
considered as shallow neural network consists of one input & output & has barely one hidden layer.	to qualify for DL there has to be atleast three layer.
Require small amounts of data	Requires large amount of unlabelled training data.
cannot perform automatic feature extraction	performs automatic feature extraction without human intervention
High performance hardware is not required.	High performance hardware is required.
Takes less time to train.	Takes lot of time to train
Input → Feature extraction → classification → output	Input → FE + classification → output

② Intelligent Agents:

- An agent can be anything that perceives environment using sensors & act upon that environment using actuators.

An agent can be:

1) Human Agent: Sensors: Eyes, Ears
   Actuators: Hand, legs
2) Robotic Agent: Sensors: cameras, IR range finder
   Actuators: Motors
3) Software Agent: Sensors: key strokes, file contents
   Actuators: Input & Display output on screen

4) Sensors:

iii) Sensors: Detects change in environment & sends information to other electronic devices.

iii) Actuators: Components of machine that convert energy into motion. Responsible for moving & controlling system. Eg: Hand, Motors.

iv) Effectors: Device that affect environment. Eg: legs, wheels etc.

⊙ Intelligent Agent

- Acts upon an environment using sensors & actuators for achieving goals.
- Eg: Thermostat

Rules for AI agent:

1) The AI agent must be able to perceive environment
2) The observation must be used to make decision.
3) The decision should result in an action.
4) The action taken by AI agent must be rational action

Structure of IA:

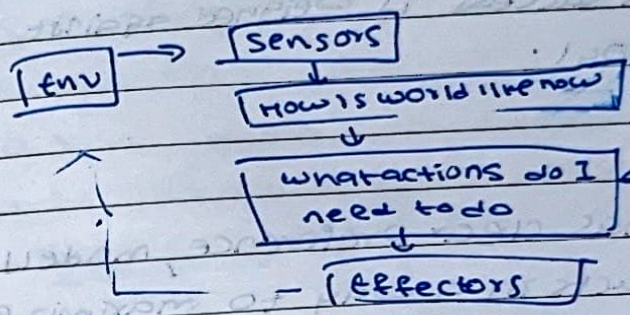
Agent = Architecture + Agent program

Architecture: machinery that agent executes on.

Agent program: implementation of agent function.

1) **Simple Reflex Agents:**

- they choose actions based on current percept.
- They are rational if correct decision is made on basis of current percept.
- environment → Completely observable
- **Condition-Action Rule**: Rule that maps state to an action



2) **Model Based Reflex Agents:**

- They use model of world to choose their actions.
- Model: knowledge of "things happen in world"
- Maintain an internal state which is a representation of unobserved aspects of current state depending on percept history.
- Updating state involves:
  - i) How the world evolves.
  - ii) How agents affect the world.

3) **Goal Based Agents:**

- They choose their action in order to achieve goals.
- They are more flexible than reflex agent since knowledge supporting decisions explicitly modeled thereby allowing for modifications.
- Goal: Description of desirable situation.

4] Utility Based Agents
- These choose their action based on utility for each state.
- Goals are inadequate when:
(i) There are conflicting goals, all of which only few can be achieved.
(ii) Goals have uncertainty of being achieved, and thus likelihood of success is weighed against importance of each goal.

5] Rational Agent:
- Agent which has clear preference, models uncertainty and acts in a way to maximize its performance measure.
- It is said to perform right things.
- Eg. used in game theory
- correct action ⇒ positive reward
- Rationality can be judged on:
(i) Performance measure which is success criterion.
(ii) Agent's prior knowledge of its environment.
(iii) Best possible actions, agents can perform.
(iv) sequence of percepts

Page No.:
Date:
YOUVA

(*) Environment:-

- PEAS is a type of model on which an AI agent works upon
- An Agent's properties can be grouped under PEAS model.
  - (P) → Performance Measure
  - (E) → Environment
  - (A) → Actuators
  - (S) → Sensor

⇒ Environment: everything in world which surrounds agent but is not part of agent itself.

(-) It is non-feministic and is where agent lives & operates.

Features:

1) Fully Observable v/s Partially Observable (no sensors → unobservable)
   Agent's current & selected action can determine its next state
2) Deterministic v/s Stochastic: Random in nature, cannot be det.

3) Episodic v/s Sequential
   - Episodic → one shot actions, only current perception required
   - Sequential → Agent requires memory of past actions to determine next one.

4) Single v/s Multi Agent
   - Single → one agent involved & working in environment
   - Multi → Multiple agents operating in an environment

5) Static v/s Dynamic
   - Static → else
     - Does not need to continue looking at world while deciding action
     - Crossword puzzle
   - Dynamic ← environment can change if agent is deliberating
     - Agents need to look at world at each action
     - Taxi driving

6) Discrete vs Continuous

↓

Finite no. of percepts & actions

Eg: Chess

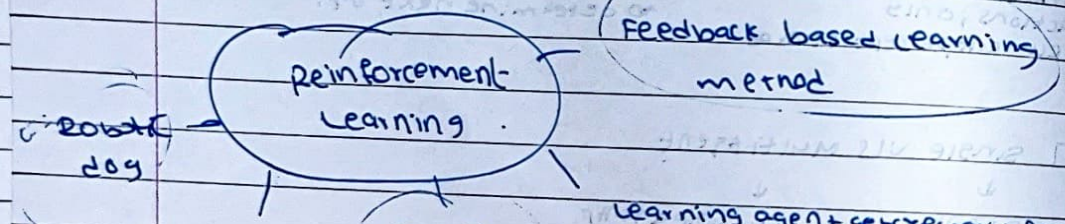
Eg. self Driving car

7) Known vs Unknown

- Results of all actions are known to agent

8) Accessible vs Inaccessible

- If agent can obtain accurate info about environment's state
- Temp. of empty room.
- Event on earth



Goal is to get most reward points

Agent learns with feedbacks & improves its performance.

Learning agent + gets reward for correct action & penalty for wrong action

Page No.:
Date:

Types of AI:

1. Reactive Machines
   - Have no memory & are task specific
   - Eg: DeepBlue that identified chess pieces & make prediction but due to no memory it cannot use past experience to inform future decision

2. Limited Memory
   - Have memory so they use past experience to inform future decisions
   - Eg: Tesla

3. Theory of Mind
   - Has social intelligence to understand emotion & predict human behavior.

4. Self Awareness
   - sense of self to understand their current state.
   - Is not existing today.

Types of ML:

1. Supervised learning
   - We provide sample labelled data
   - On this basis it predicts output & its goal is to map input data to output data
   - Based on supervision
   - Eg: Spam filtering

2. Unsupervised Learning
   - Machine learns without supervision
   - We provide unlabelled sample data
   - Goal is to restructure input data in new features

U2 :

Hill Climbing

1) Local Maximum:
Solution: Backtracking. Create list of promising paths so algo can backtrack search & explore other paths as well.

2) Plateau :
Solution: Take big or very little steps while searching. Randomly select state which is far away from current state

3) Ridges: Higher than surrounding areas but itself has slope.
Solution: With the use of bidirectional search or by moving in different region.

Heuristic Function: (Tourguide)

- Function in informed search as it finds the most promising path.
- Takes current state of agent as input & estimates of how close agent is from goal.
- Might not always give best solution but is guaranteed to provide good soln in reasonable time.

$h(n) \leq h^*(n)$

↓ Heuristic cost ↓ Estimated cost