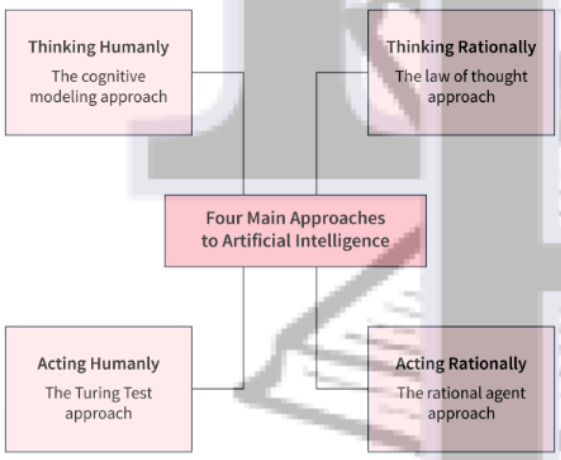
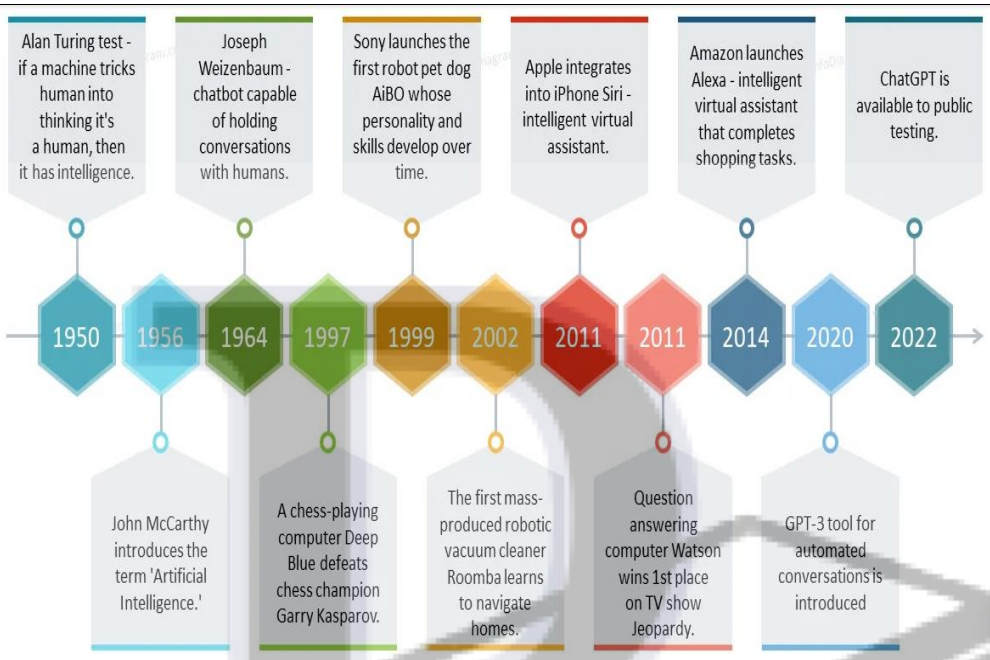
School of Computer Science and Engineering
Winter Semester 2023-24
Continuous Assessment Test – I - KEY
SLOT: B1+TB1

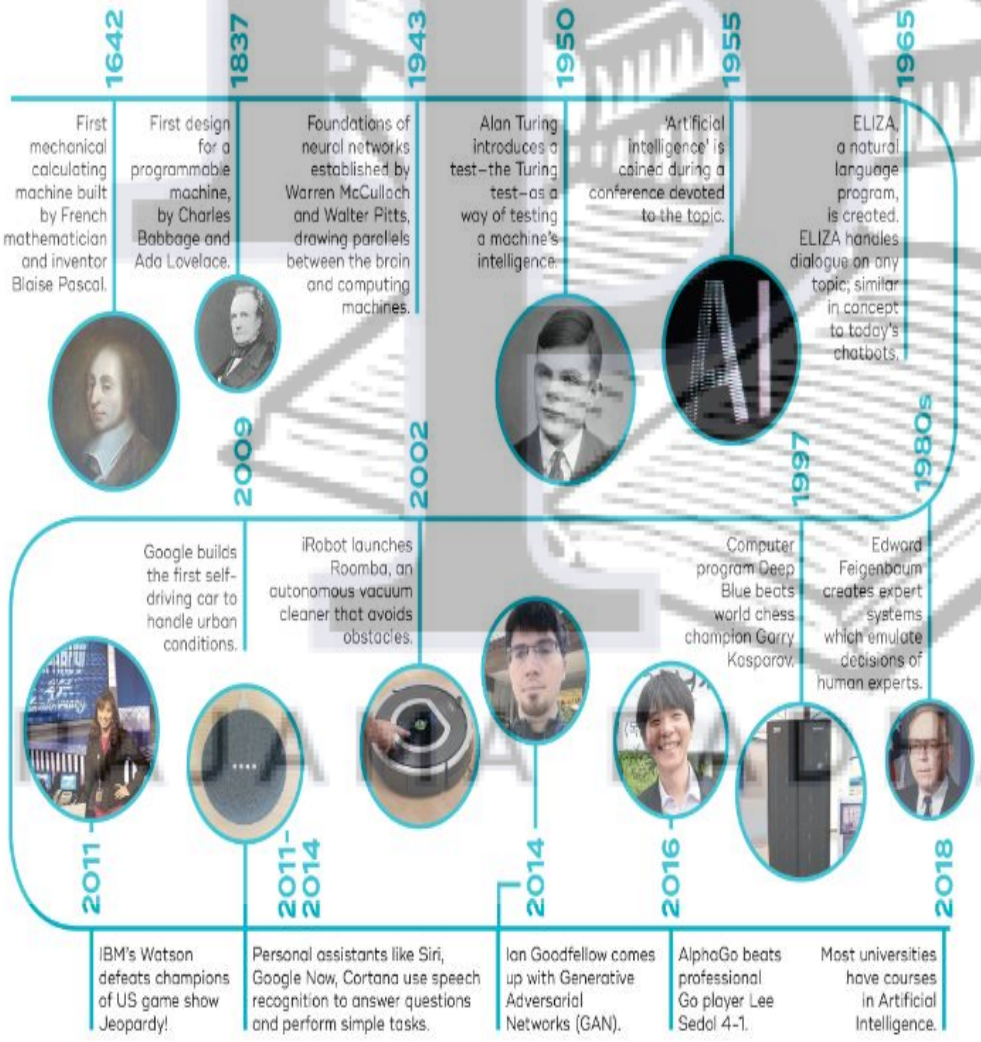
Programme Name & Branch: B.Tech. & Computer Science & Engineering / SCOPE
Course Name & Code: Artificial Intelligence & BCSE306L
Class Number (s): Common for all batches **Faculty Name (s):** All
Exam Duration: 90 Min. **Maximum Marks: 5*10 = 50**

Q. No.	Question (Answer all questions)	Max Marks	CO	BL
1.	a) Describe different approaches to Artificial Intelligence. List the timeline of the evolution of artificial intelligence and briefly describe each period/breakthrough. [5 M] b) Consider an artificial agent learning to play chess, where the agent learns the game's rules and optimal moves through multiple plays and feedback from critics. Which type of agent would be most suitable for a chess-playing agent? Justify your answer. Also, briefly describe the agent architecture with a suitable diagram. [5 M]	10	CO 1	BL 3
Answer	**a)** Following are the four different approaches to Artificial Intelligence **[2 Marks]** • 1 Mark for naming four approaches + 1 Mark for brief explanation of each category.  **Timeline of the evolution of artificial intelligence [3 Marks]** **Sample Solution 1:**			



Source link: https://www.infodiagram.com/slides/ai-development-timeline/

Sample solution 2:



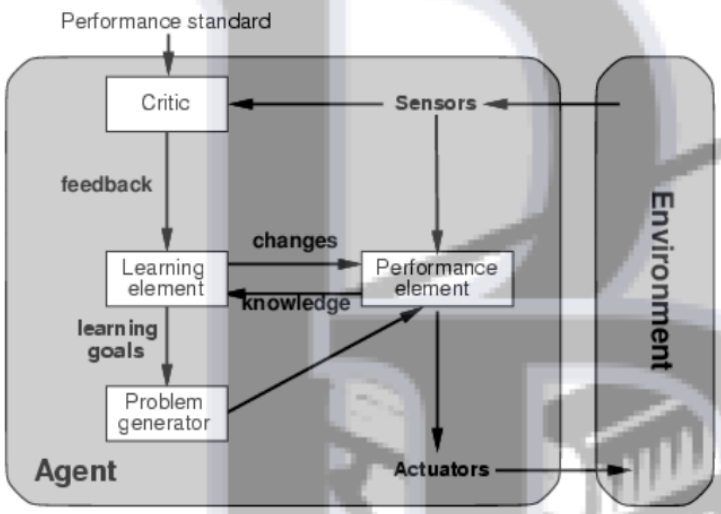
Source: https://digitalwellbeing.org/artificial-intelligence-timeline-infographic-from-eliza-to-tay-and-beyond/

Othe similar timeline:

- https://www.linkedin.com/pulse/artificial-intelligence-timeline-2021-dr-mansoor-agha-siddiqui/
- https://qbi.uq.edu.au/brain/intelligent-machines/history-artificial-intelligence
- https://digitalwellbeing.org/artificial-intelligence-timeline-infographic-from-eliza-to-tay-and-beyond/

b) Learning based Agent is most suitable as the agent is game's rules and optimal moves through multiple plays and feedback from critics. **[2 Marks]**

Architecture of Learning based Agent [2 Marks]



Following are the major components of Leanring based Agent: [1 Marks]

i. **Learning element** is responsible for making improvements.
ii. **Performance element** is responsible for selecting external actions.
iii. **Critic** provides feedback on how the agent is doing and determines how the performance element should be modified to do better in the future.
iv. **Problem generator** is responsible for suggesting actions that will lead to new and informative experiences.

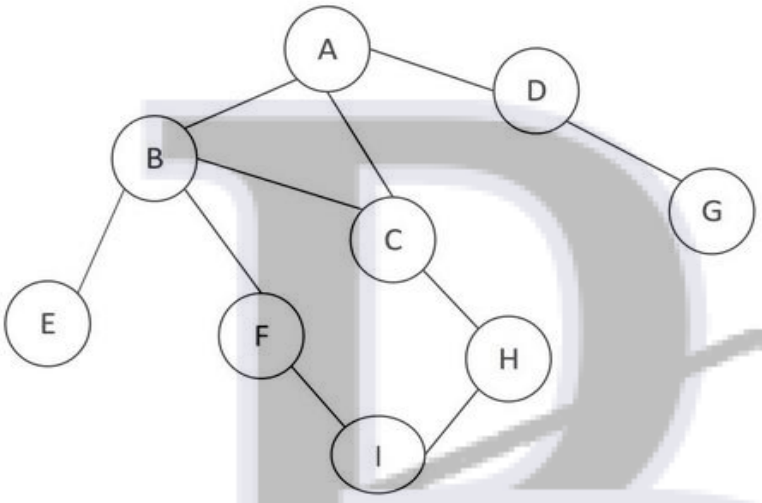
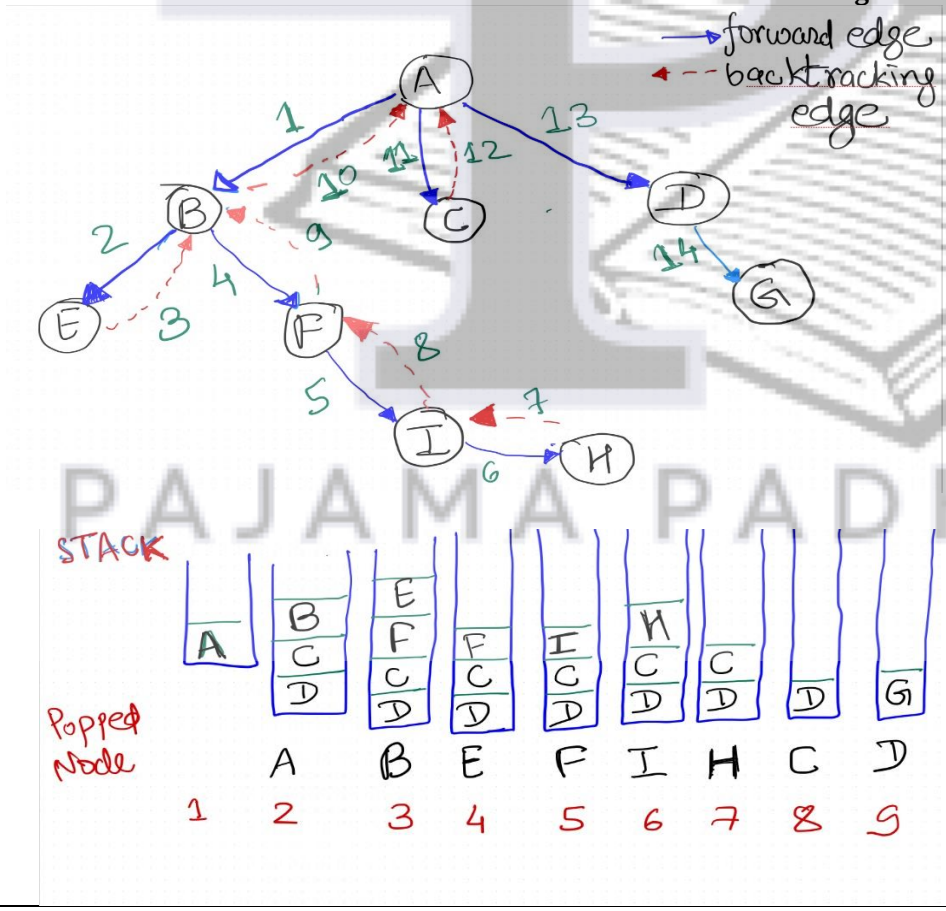
Note: Partial marks may be provided for giving other agent types and its architecture diagram based on justification provided by student.

2.	Describe the task environments and their characteristics for the following agents. i. Rental Bike/Car booking ii. Cooking Robot iii. Grocery Delivery iv. Playing Chess	10	CO 1	BL 4

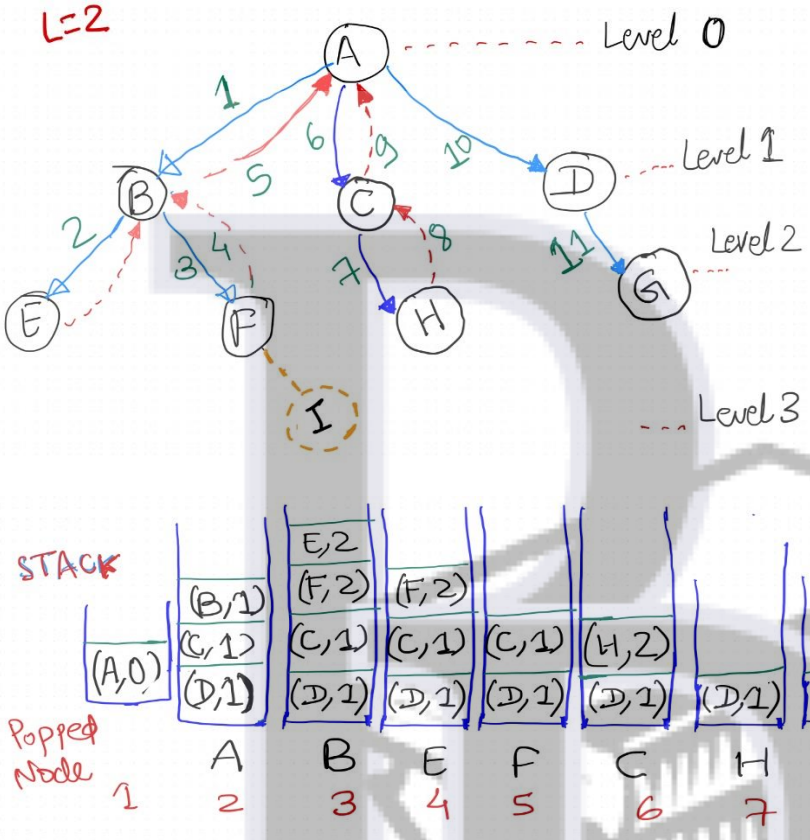
Answer

The characteristics of task environment are as follows**: [4*2.5 = 10 M]**

Task Environment	Observable	Agent	Deterministic	Episodic	Static	Discrete
i) Rental Bike/Car booking	Partially	Multi	Non-Deterministic	Episodic	Dynamic	Continuous
ii) Cooking Robot	Fully	Single	Deterministic	Sequential	Static	Discrete
iii) Grocery Delivery	Partially	Multi	Non-Deterministic	Episodic	Dynamic	Discrete
iv) Playing Chess	Fully Observable	Multi	Non-Deterministic	Sequential	static	Discrete

3.	Apply the following search algorithms to find a path from the node A to G in the given graph: [3 * 2 = 6 M] i) Depth First Search (DFS) ii) Depth-limited search (L=2) iii) Iterative Deepening Search (IDS)  **Note:** Process the nodes alphabetically for the nodes at the same level. b) Analyze the above algorithms regarding completeness, optimality, time complexity, and space complexity. [4 M]	10	CO 2	BL 5
Answer	**a) i)Depth First Search [3 Marks]** For a popped-out node (*v*) from STACK, we are inserting unvisited neighbor of popped node *v* in the STACK in descending order of alphabets. • **1.5 Marks for stack and 1.5 marks for DFS traversal diagram**  			

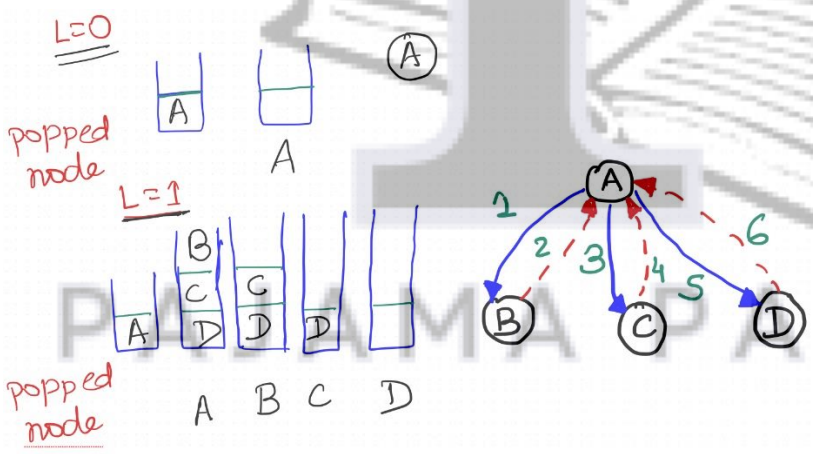
ii) Depth-limited search (L=2) **[3 Marks]**

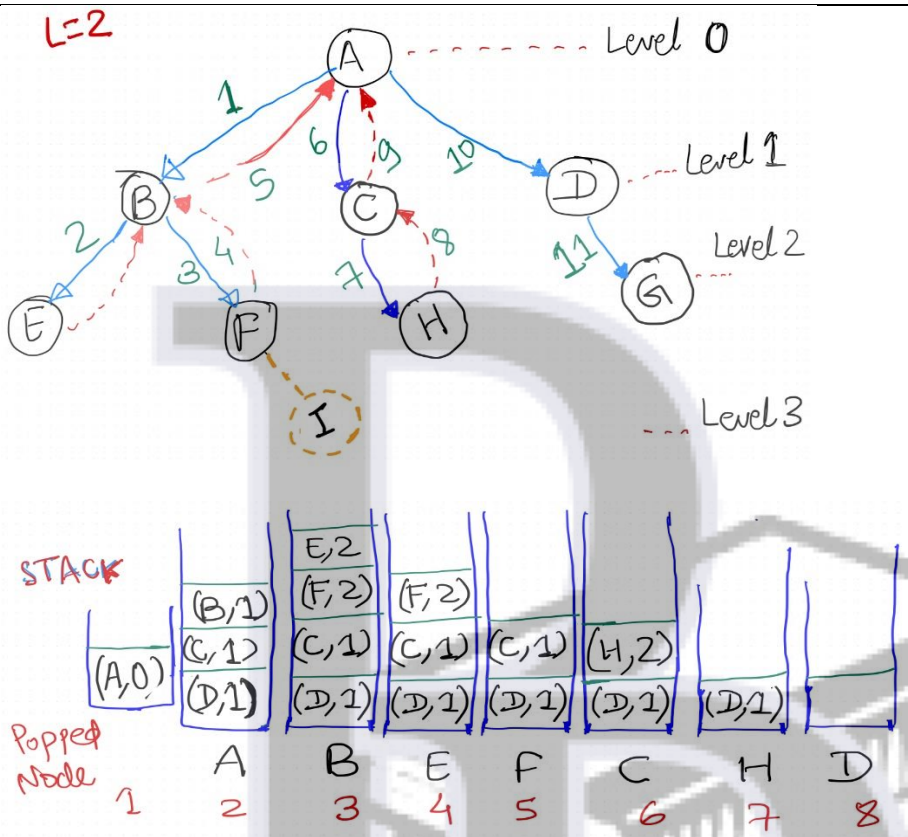


Neigh(D) = G; Found Node G

Note:- Here, depth of node is also inserted with node in stack.

iii) Iterative Deepening Search **[3 Marks]**





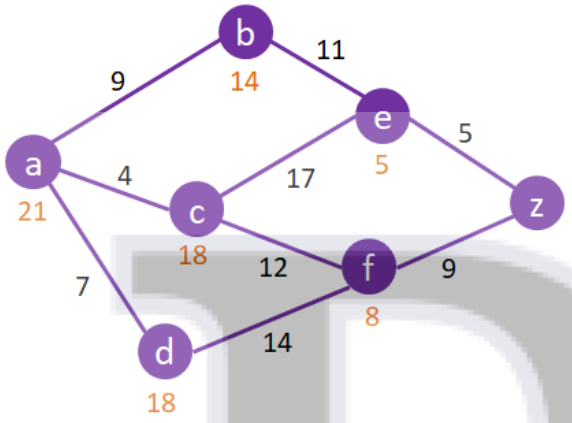
(Neigh(D) = G; Found Node G

Note:- Here, depth of node is also inserted with node in stack.

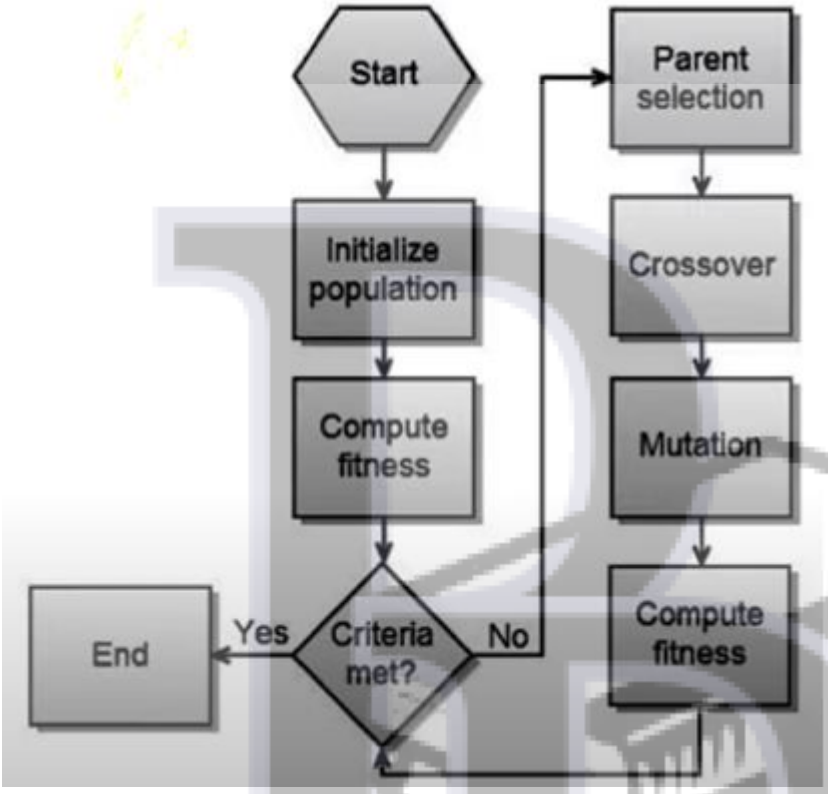
b) Anlaysis of Algorithm

Algorithm Name	**Completenes**	**Optimality**	**Time Complexity**	**Space Complexity**
Depth First Search (DFS)	No	No	$O(bm)$	$O(bm)$
Depth Limited DFS	No	No	$O(b^l)$	$O(bl)$
Iterative Deepening DFS	Yes	Yes	$O(bd)$ when there is solution $O(bm)$ when there is no solution	$O(bd)$ when there is solution $O(bm)$ when there is no solution

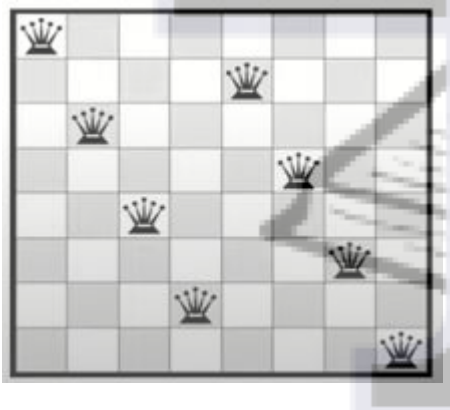
where
b is branching factor
m is the maximum depth of tree
d is the depth where the goal node v is found.

4.	Apply A* Search algorithm to find the shortest path from a to z using the following graph:  The heuristic values for each node is given below: **Node** | a | b | c | d | e | f **h(n)** | 21 | 14 | 18 | 18 | 5 | 8	10	CO 2	BL 5
Answer	We will traverse the graph based on the f(n) Where f(n) = g(n) + h(n); g(n) = cost to reach node n, h(n) = heuristic cost to reach goal from node n **Nodes** | **Extracted node** **S. No** | **a** | **b** | **c** | **d** | **e** | **f** | **z** | **Node** | ***f(n)*** 21 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | a | 0 0 | 0+9+14 = 23 | 0+4+18 = **22** | 0+7+18 = 25 | ∞ | ∞ | ∞ | c | 0+4 =4 0 | **23** | 4 | **25** | 4+17+5 = 26 | 4+12+8 = 24 | ∞ | b | 0+9 = 9 0 | 9 | 4 | **25** | 9+11+5 = 25 | 24 | ∞ | f | 16 0 | 9 | 4 | 25 | 9+11+5 = 25 | 16 | 25 | z | 25 Hence path cost from node a->z is **25** and the path is **a->c->f->z.**			
5.	Explain each step of the genetic algorithm with correct terminology in detail using the eight-queen problem and appropriate examples. Calculations must be provided along with the steps when necessary.	10	CO 2	BL 6
Answer	**Genetic Algorithm**: Genetic algorithm is a heuristic search algorithm inspired from Darwin's theory of Natural evolution. In Genetic algorithm, we **encode** each state of a search problem with a string (binary/decimal/hexadecimal/alphabets etc.). We initialize with a population of individuals. Further, we define a **fitness function** to estimate how good an individual is w.r.t. to target goals. Using fitness function, we select the group parents to participate in production next generation of individuals using crossover and mutation. We keep on repeating the process until the stopping criteria is met.			

The flowchart of Genetic Algorithm can be illustrated using the diagram given below.



8 - Queen problem: Arrange the 8 queens in 8*8 Chess board such that no queens attach each other (horizontally, vertically, or diagonally).

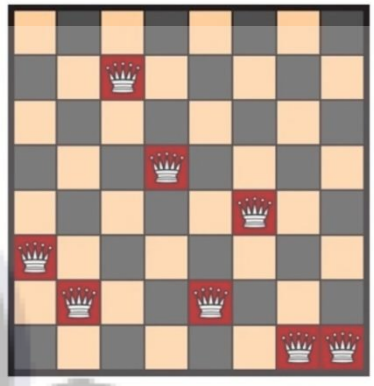


With respect to solving 8 Queen problems using Genetic algorithm, we will demonstrate following steps:

i. Define Encoding mechanism: Representing an individual.
ii. Define Fitness Function: to estimate how good an individual of a population is.
iii. Initialize a population.
iv. Calculate the Fitness of all individuals of the population using fitness function.
v. Selection of parent for generating new child.
vi. Cross of selected parents
vii. Mutation of selected parents
viii. Repeat step (iv) to (vii) until the stopping criteria is met.

i. Defining Encoding Mechanism for 8 Queen problem

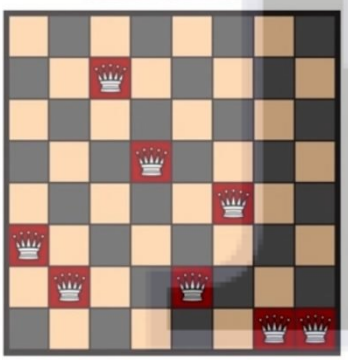
- Formulate an appropriate method to represent individuals of a population.
- Array.
- Index: Column.
- Value: Row.



3	2	7	5	2	4	1	1

ii. Define a fitness function.

Individual



3	2	7	5	2	4	1	1

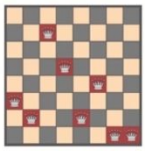
Fitness = No. of non attacking pairs

- **Queen 1:** 6
- **Queen 2:** 5
- **Queen 3:** 4
- **Queen 4:** 3
- **Queen 5:** 3
- **Queen 6:** 2
- **Queen 7:** 0
- **Queen 8:** 0
- **Total**: 23

iii. Initialize a population (assume population size =4)

- Generate random arrangements of 8 queens on a standard chess board.

A



3	2	7	5	2	4	1	1

B

2	4	7	4	8	5	5	2

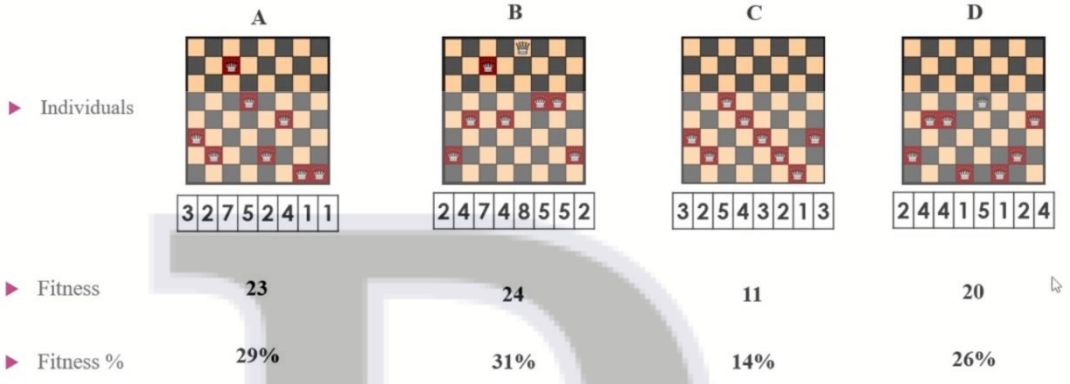
C

3	2	5	4	3	2	1	3

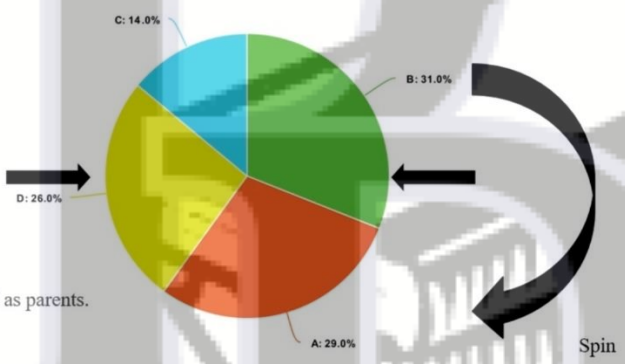
D

2	4	4	1	5	1	2	4

iv. Calculate the Fitness of all individuals of the population using fitness function.

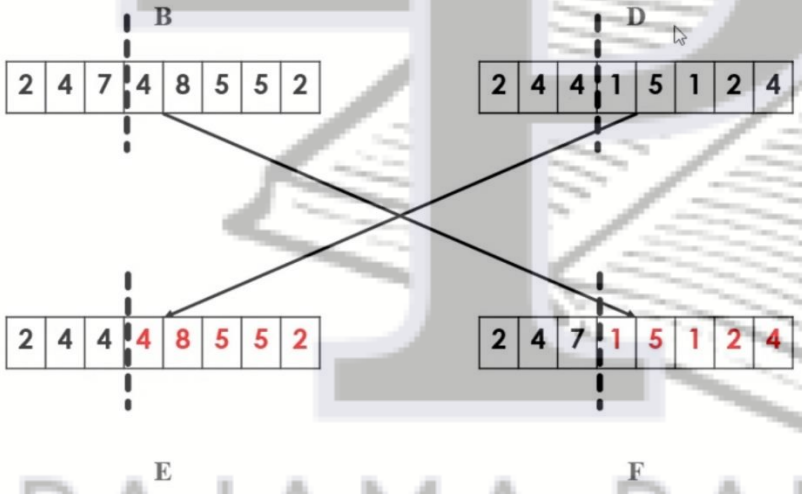


v. Selection of parent for generation of new child

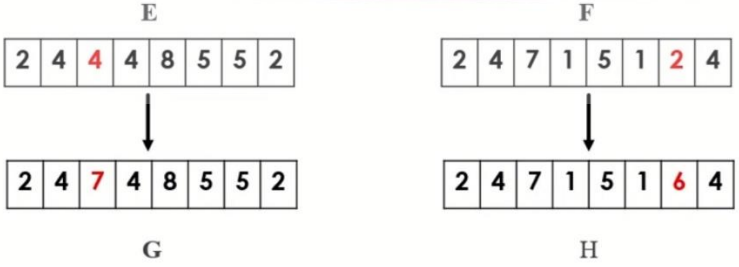


- B and D are selected as parents.

vi. Crossover between selected parents



vii. Mutation of selected parents



viii. Stopping Criteria Steps (iv) to (vii) are repeated until highest fitness score (28 in this case) is achieved