Question 1.
Exercise Search Process : Define problem formulation for Farmer, wolf, chicken, grain
A farmer is on one side of a river and wishes to cross the river with a wolf, a chicken, and a bag of grain. He can only take one item at a time in his boat with him. He must be in the boat in order for it to cross the river. He cannot leave the chicken alone with the grain, or it will eat the grain. (Note: the chicken will not eat the grain if the farmer is on that side of the river, either in the boat or on shore.) He cannot leave the wolf alone with the chicken or the wolf will eat the chicken. (Note: the wolf will not eat the chicken if the farmer is on that side of the river, either in the boat or on shore.) The farmer's problem is to get all three safely across the river to the other side.

Answer

A. Provide a precise problem formulation that includes an initial state, possible actions and a goal test. Assume a unit path cost.

State Representation

0F → represents the farmer's Left side.
1F → represents the farmer's Right side.
0W → represents the wolf's Left side.
1W → represents the wolf's Right side.
0C → represents the chicken's Left side.
1C → represents the chicken's Right side.
0G → represents the grain's Left side.
1G → represents the grain's Right side.
0B → represents the boat's Left side.
1B → represents the boat's Right side.

An Initial State

(0F , 0W , 0C , 0G , 0B) → farmer , wolf , chicken , grain , and boat are all on the starting Left side.

Goal State / Goal Test

(1F , 1W , 1C , 1G , 1B) → farmer , wolf , chicken , grain and boat are all on the opposite Right side.

Possible Actions

1. Farmer with chicken
2. Farmer alone
3. Farmer with grain
4. Farmer with wolf

A unit path cost
Each crossing of the river has a unit cost of 1.

B. Provide the optimal solution to this problem.

Method - 1

Left Bank of River		Right Bank of River
Initial State **(0F , 0W , 0C , 0G , 0B)**		( )
(0W , 0G)	→ (+1) **(0F , 0C , 0B)** **Farmer with chicken**	**(1F , 1C , 1B)**
(0F , 0W , 0G , 0B)	← (-1) **(1F , 1B)** **Farmer alone**	**(1C)**
(0W)	→ (+1) **(0F , 0G , 0B)** **Farmer with grain**	**(1F , 1C , 1G , 1B)**
(0F , 0W , 0C , 0B)	← (-1) **(1F , 1C , 1B)** **Farmer with chicken**	**(1G)**
(0C)	→ (+1) **(0F , 0W ,0B)** **Farmer with wolf**	**(1F , 1W , 1G , 1B)**
(0F , 0C , 0B)	← (-1) **(1F , 1B)** **Farmer alone**	**(1W , 1G)**
( )	→ (+1) **(0F , 0C , 0B)** **Farmer with chicken**	**Goal state** **(1F , 1W , 1C , 1G , 1B)**

Method - 2

Left Bank of River		Right Bank of River
Initial State (0F , 0W , 0C , 0G , 0B)		( )
(0W , 0G)	→ (+1) (0F , 0C , 0B) Farmer with chicken	(1F , 1C , 1B)
(0F , 0W , 0G , 0B)	← (-1) (1F , 1B) Farmer alone	(1C)
(0G)	→ (+1) (0F , 0W , 0B) Farmer with wolf	(1F , 1C , 1W , 1B)
(0F , 0G , 0C , 0B)	← (-1) (1F , 1C , 1B) Farmer with chicken	(1W)
(0C)	→ (+1) (0F , 0G ,0B) Farmer with grain	(1F , 1W , 1G , 1B)
(0F , 0C , 0B)	← (-1) (1F , 1B) Farmer alone	(1W , 1G)
( )	→ (+1) (0F , 0C , 0B) Farmer with chicken	Goal state (1F , 1W , 1C , 1G , 1B)

C. **Which of the uninformed search strategies will prove to be the best to apply to this problem?**

Iterative Deepening Depth First Search for the Farmer's Problem.

The Iterative Deepening Depth First Search would be an effective uninformed search strategies.

1. Iterative Deepening Depth First Search is complete , meaning it will always find a solution if one exists. The farmer problem because we need to ensure that we find a solution that safely moves all items and the farmer across the river.
2. Iterative Deepening Depth First Search is optimal solution, since the state space is small
   and given that each step has the same cost. The search space in a depth first manner while iteratively increasing the depth limit, ensuring that the shortest path to the goal state is found.
3. The modest memory is farmer, Iterative Deepening Depth First Search is memory efficient compared to other search algorithms like breadth-first search. It only requires enough memory to store the current path being explored, which makes it suitable for problems with large or infinite state spaces.