Chapter – 2
Problem Solving

Bal Krishna Nyaupane

Assistant Professor

Department of Electronics and Computer Engineering

IOE,TU

bkn@pcampus.edu.np

Introduction to Problem solving

- Problem solving is a agent based system that finds sequence of actions that lead to desirable states from the initial state.
- ***Four steps of problem solving are:***
  1. **Goal Formulation:** Helps to organize behavior by isolating and representing the task knowledge necessary to solve problem.
  2. **Problem Formulation:** Problem formulation is the process of deciding what actions and states to consider, and follows goal formulation. It define the problem precisely with initial states, final state and acceptable solutions.
  3. **Searching:** Determine the possible sequence of actions that lead to the states of known values and then choosing the best sequence.
  4. **Execution:** Once the search algorithm returns a solution to the problem, the solution is then executed by the agent.

Problem formulation

- Problem formulation is the process of deciding what actions and states to consider, given a goal. A problem is defined by:
  - **An initial state:** State from which agent start
  - **Successor function:** The set of possible actions available to the agent. The term operator is used to denote the description of an action in terms of which state will be reached by carrying out the action in a particular state.
  - **Goal test:** Determine whether the given state is goal state or not . For example, in the eight-puzzle there is a single solution and a ***single goal state***, where as in chess there are many winning positions and hence ***many goal states***.
  - **Path cost:** Sum of cost of each path from initial state to the given state.
- A problem is when defined with these components is called **Well Defined Problem.**

State Space Representation

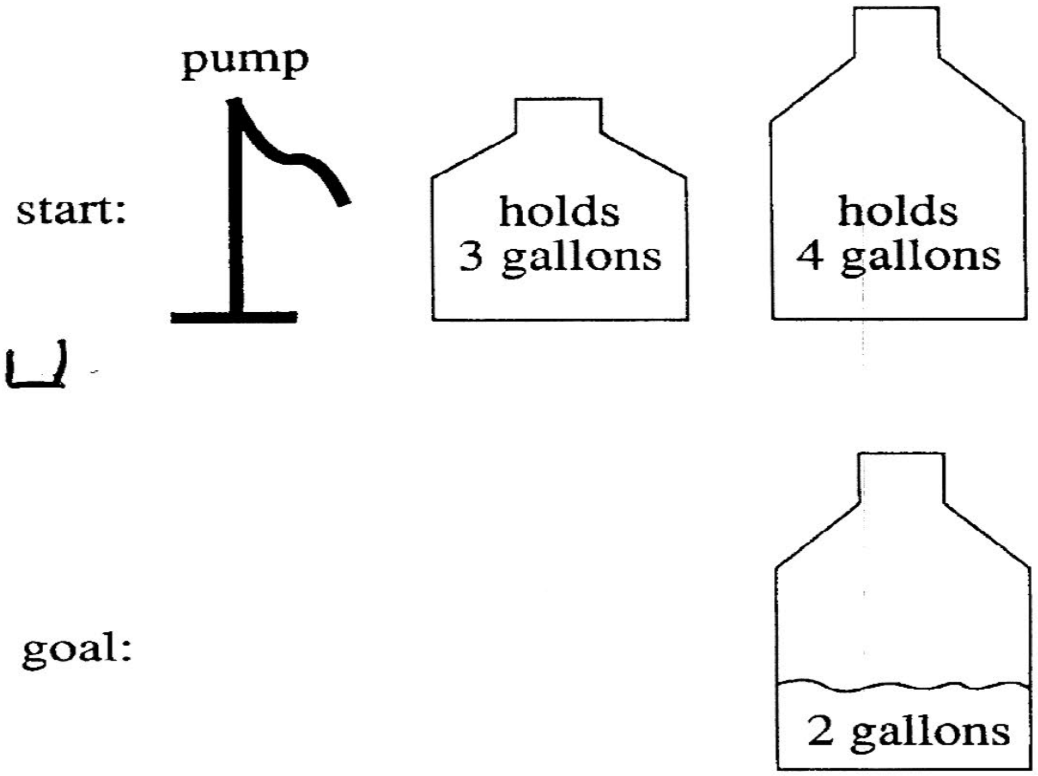
- The state space is commonly defined as a directed graph in which each node is a state and each arc represents the application of an operator transforming a state to a successor state.
- A state space essentially consists of a set of nodes representing each state of the problem, arcs between nodes representing the legal moves from one state to another, an initial state and a goal state.
- In state space search is formally represented as
  - ***A set of states****;*
  - ***Initial state*** *- where the agent is ;*
  - ***Operator set*** *- set of action to go from one state into another ;*
  - *A set of* ***Goal states;***
  - ***Path coast*** *- the cost of the path from the initial to the goal state.*
- **State space -** the set of states generated by the set of operators .

Formalizing a Problem

- **Basic steps:**
  1. Define a state space
  2. Specify one or more initial states
  3. Specify one or more goal states
  4. Specify a set of operators that describe the actions available:
     - What are the unstated assumptions?
     - How general should the operators be made?
     - How much of the work required to solve the problem should be represented in the rules?

Water Jug Problem

- A Water Jug Problem: You are given two jugs, a **4-gallon** one and a **3-gallon** one, and a pump which has unlimited water. How can you get exactly **2 gallons of water in the 4-gallon jug**?



- **State Representation** :(x, y)
  - x: Contents of four gallon; $0 \leq x \leq 4$
  - y: Contents of three gallon; $0 \leq y \leq 3$
- **Initial state: (0,0)**
- **Goal state** : (2,y) where $0 \leq y \leq 3$.
- **Operators:**
  - Fill 3-gallon from pump, fill 4-gallon from pump
  - Fill 3-gallon from 4-gallon , fill 4-gallon from 3-gallon
  - Empty 3-gallon into 4-gallon, empty 4-gallon into 3-gallon
  - Dump 3-gallon down drain, dump 4-gallon down drain

Production Rules for the Water Jug Problem

1) If $x < 4$, Fill the 4-gallon jug ===> $(x,y) \rightarrow (4,y)$
2) If $y < 3$, Fill the 4-gallon jug ===> $(x,y) \rightarrow (x,3)$
3) If $x > 0$, Empty the 4-gallon jug on the ground ===> $(x,y) \rightarrow (0,y)$
4) If $y > 0$, Empty the 3-gallon jug on the ground ===> $(x,y) \rightarrow (x,0)$

5) If $x + y \geq 4$ and $y > 0$, pour water from the 3-gallon jug into the 4-gallon jug until the 4-gallon jug is full ===> $(x,y) \rightarrow (4, y - (4 - x))$

6) If $x + y \geq 3$ and $x > 0$, pour water from the 3-gallon jug into the 4-gallon jug until the 4-gallon jug is full ===> $(x,y) \rightarrow (x - (3 - y), 3)$

7) If $x + y \leq 4$ and $y > 0$, pour all the water from the 3-gallon jug into the 4-gallon jug ===> $(x,y) \rightarrow (x + y, 0)$

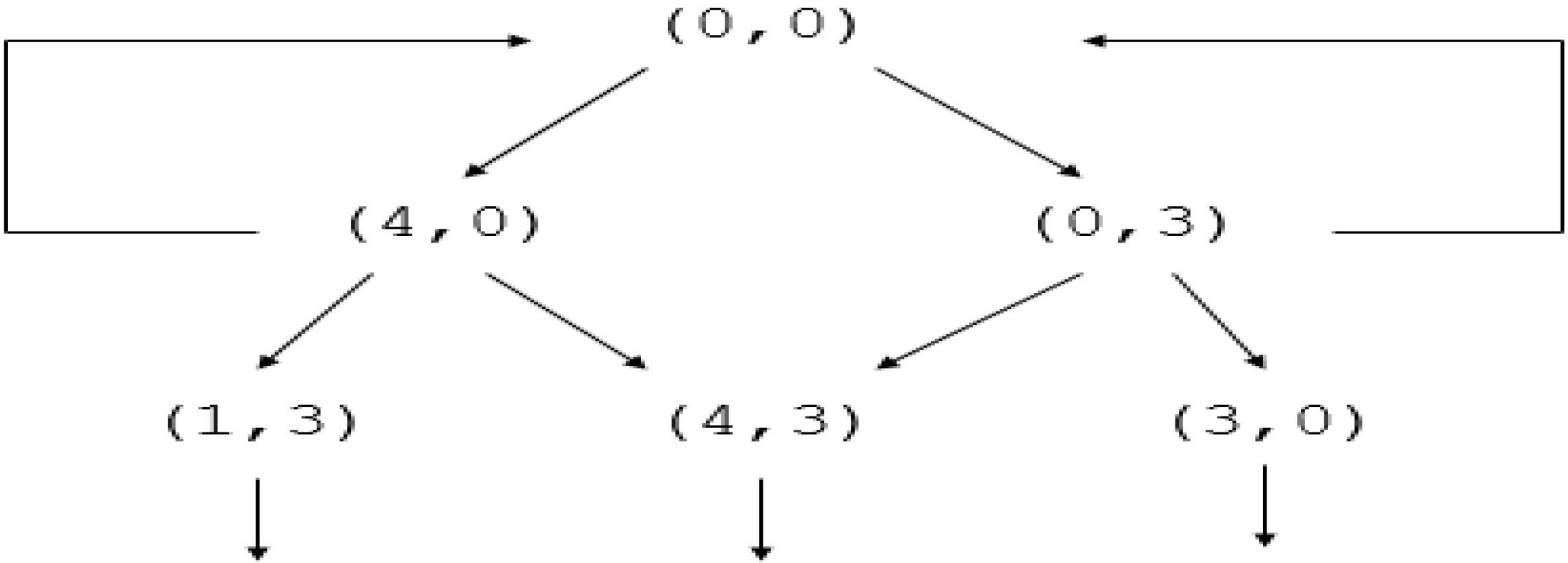
8) If $x + y \leq 3$ and $x > 0$, pour all the water from the 4-gallon jug into the 3-gallon jug ===> $(x,y) \rightarrow (0, x + y)$

Two Different Solutions to the Water Jug Problem

Gallons in the 4-Gallon Jug	Gallons in the 3-Gallon Jug	Rule Applied
0	0	Initial State
0	3	2
3	0	7
3	3	2
4	2	5
0	2	3
2	0	7

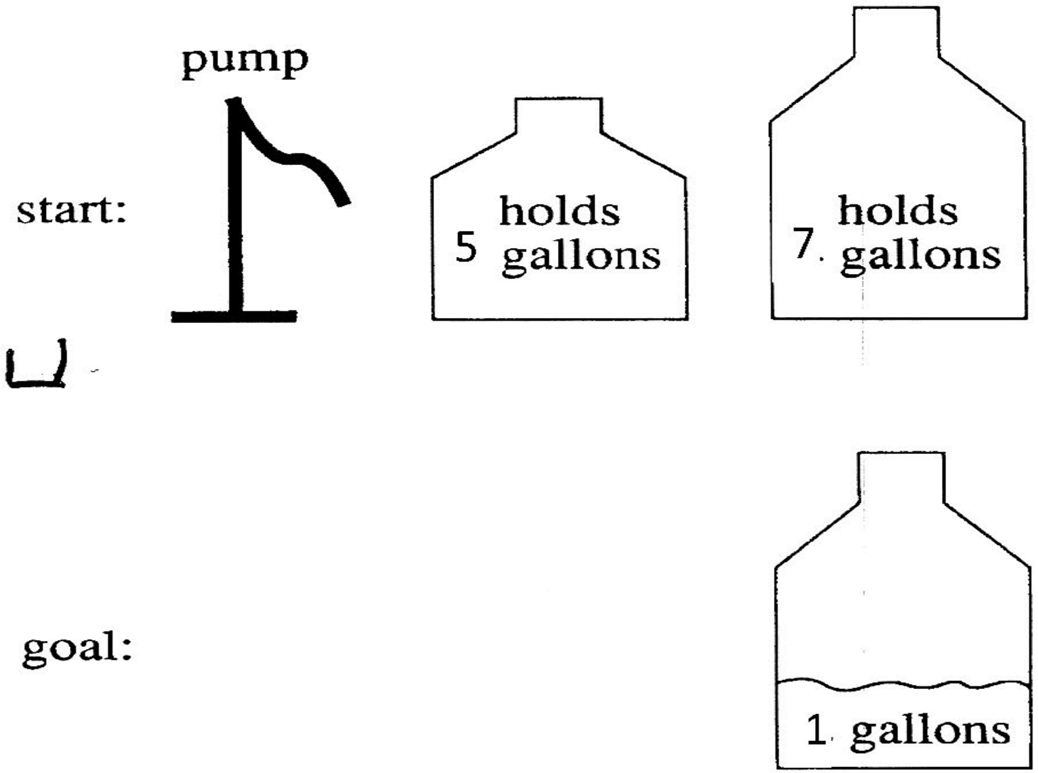
Gallons in the 4-Gallon Jug	Gallons in the 3-Gallon Jug	Rule Applied
0	0	Initial State
4	0	1
1	3	6
1	0	4
0	1	8
4	1	1
2	3	6

Search tree for Water Jug Problem



Water Jug Problem

- A Water Jug Problem: You are given two jugs, a **5-gallon** one and a **7-gallon** one, and a pump which has unlimited water. How can you get exactly 1 **gallons of water in the 7-gallon jug**?



- **State Representation** :(x, y)
  - x: Contents of four gallon; $0 \le x \le 5$
  - y: Contents of three gallon; $0 \le y \le 7$
- **Initial state: (0,0).**
- **Goal state** : (0,1).

8- Puzzle Problem

- ***States:*** A state description specifies the location of each of the eight tiles in one of the nine squares. For efficiency, it is useful to include the location of the blank.
- ***Successor Function:*** Generates the legal states from trying the four actions {*Left, Right, Up, Down*}
- ***Goal Test:*** Checks whether the state matches the goal configuration
- ***Path Cost:*** Each step costs 1, so the path cost is just the length of the path

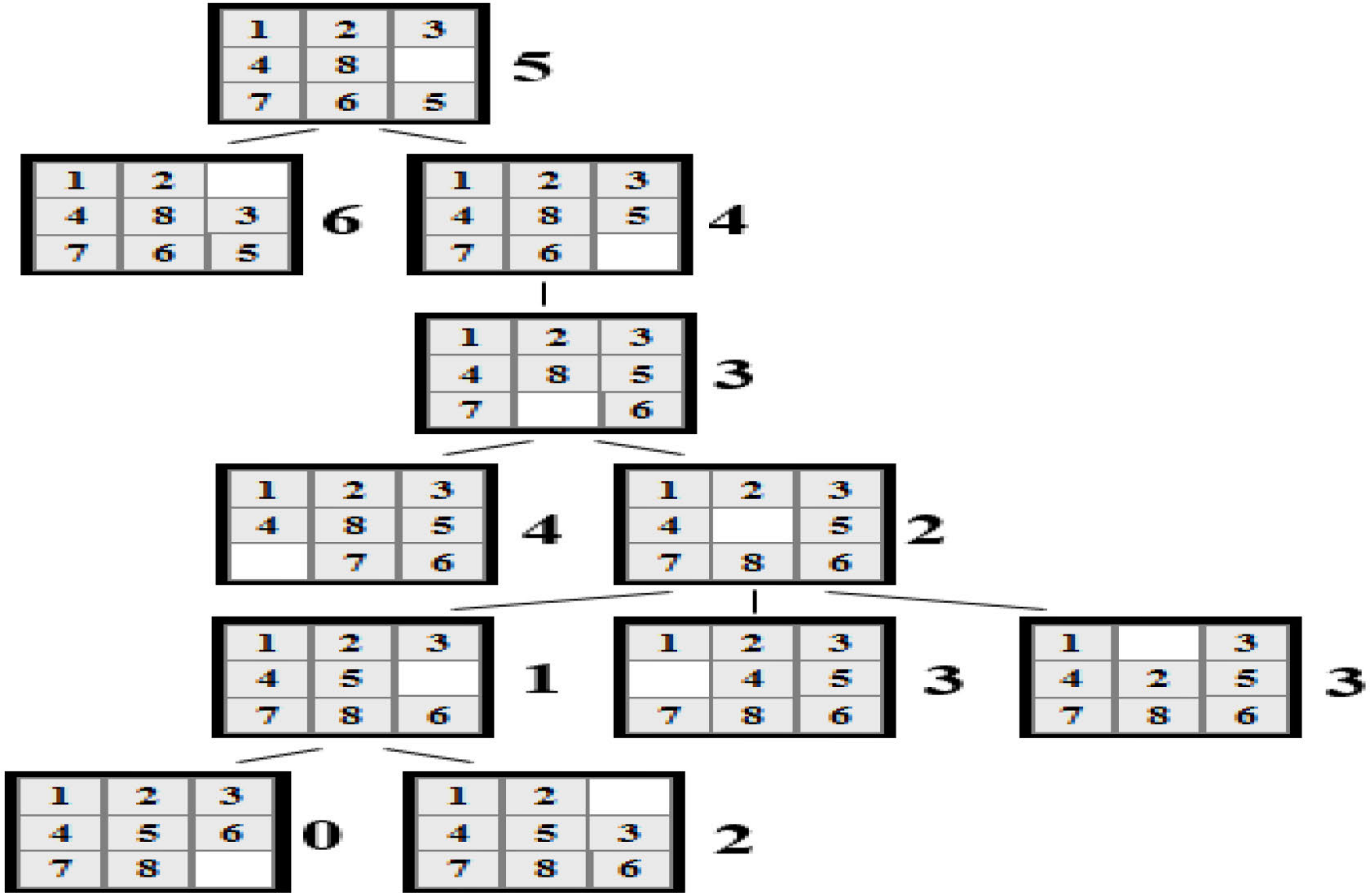
1	**2**	**3**
4	**8**	
7	**6**	**5**

Initial State

1	**2**	**3**
4	**5**	**6**
7	**8**	

Goal State

Search Tree for 8- Puzzle problem



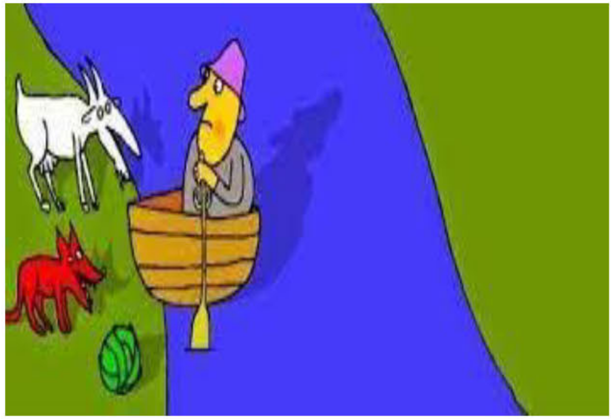
Farmer, wolf, goat, and corn problem

- A farmer with his wolf, goat, and Corn come to the West Side of a river. How wants to get all of his animal and his Corn across the river onto the East Side. There is a boat at the river's edge, but, of course, only the farmer can row. The boat also can carry only two things, including the rower, at a time. If the wolf is ever left alone with the goat, the wolf will eat the goat; similarly if the goat is left alone with the Corn, the goat will eat the Corn. How can the farmer get everything on the East Side?

  i. *Formulate this puzzle as search*

  ii. *Solve this problem using search*

  iii. *Draw the search tree and show the final solution.*



Farmer, wolf, goat, and corn problem

- **Representation**
  - ***State(F, W, G, C)*** describes the location of Farmer, Wolf, Goat, and Corn
  - ***Path cost:*** number of crossings made.
  - ***Initial state*** is state **F W D C**: everyone is on the West bank
  - ***Goal state*** is state ________ : everyone is on the East bank
  - ***Set of Operators:*** In each move, the farmer crosses the river with wither the wolf, the goat, the corn, or nothing. Each move can be represented with a corresponding atom: wolf, goat, corn, and nothing.
- For example:

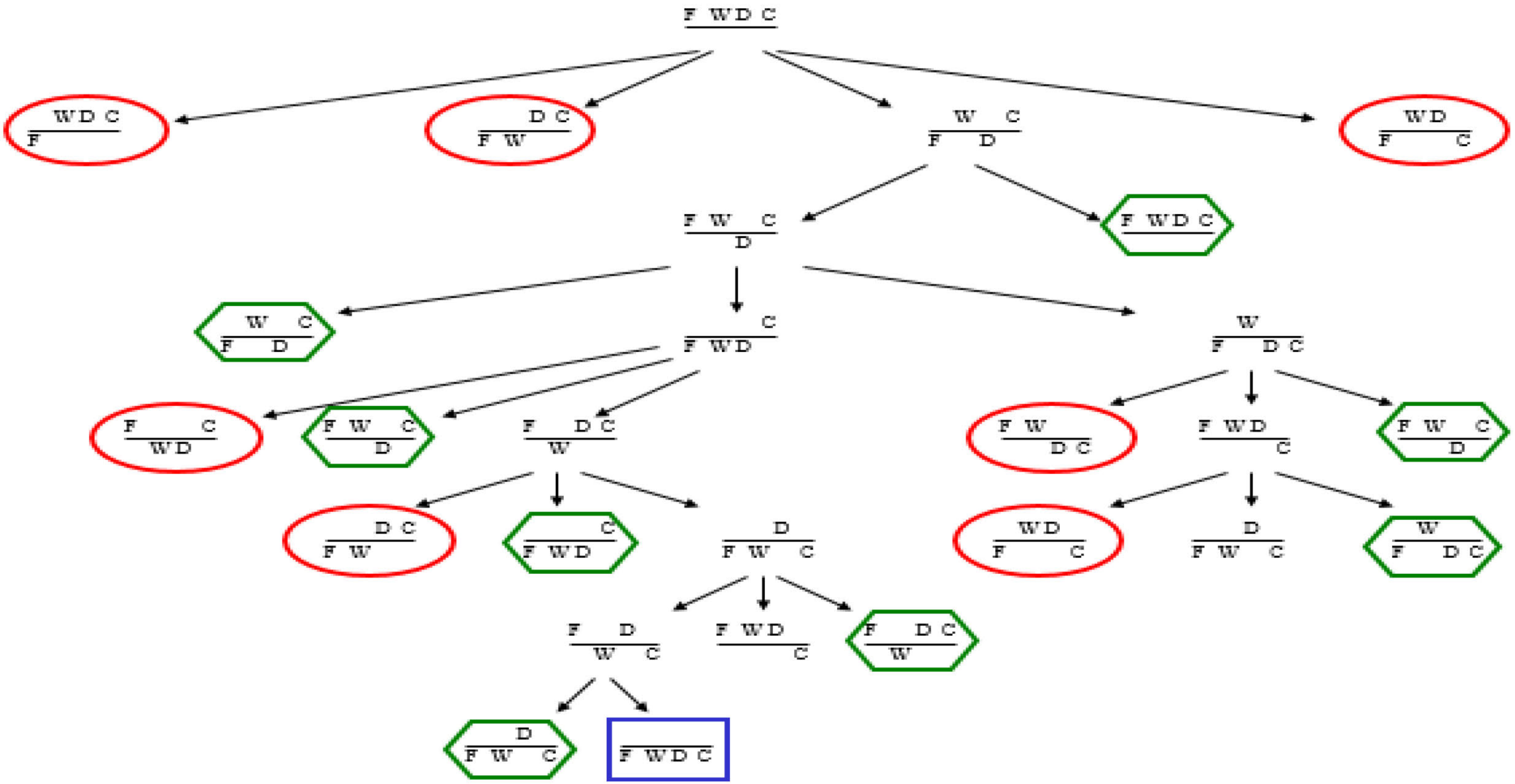
This means that everybody/everything is on the same side of the river.

F W D C

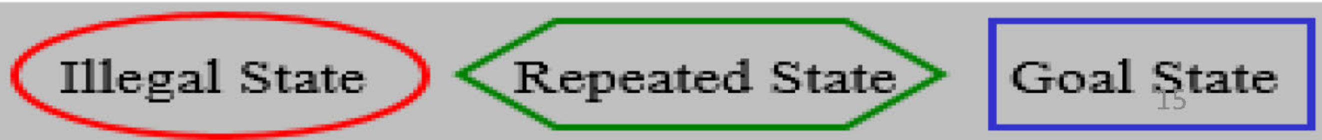
This means that we somehow got the Wolf to the other side.

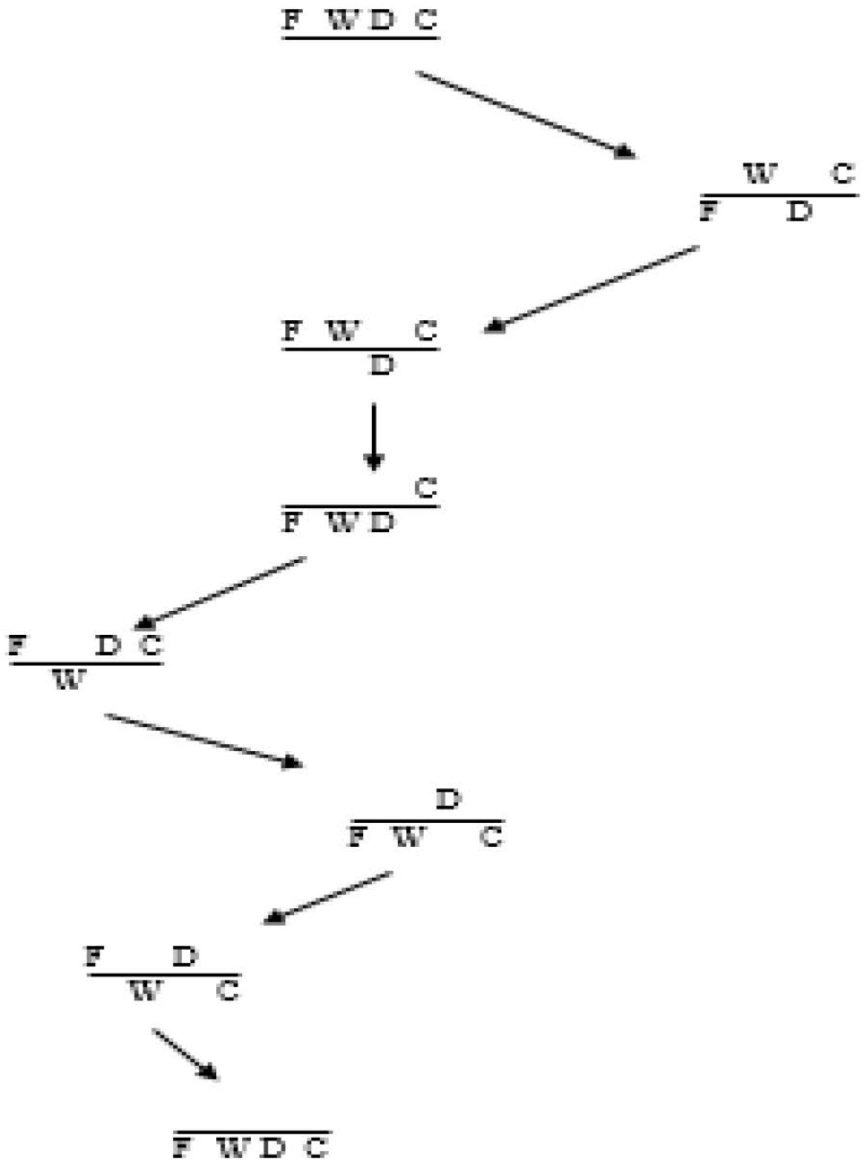
F D C

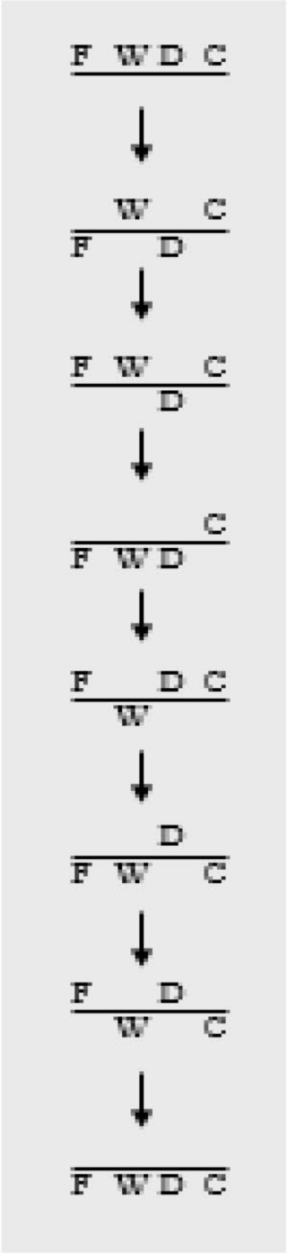
W



Search Tree for "Farmer, Wolf, Duck, Corn"







Initial State

Farmer takes duck to left bank

Farmer returns alone

Farmer takes wolf to left bank

Farmer returns with duck

Farmer takes corn to left bank

Farmer returns alone

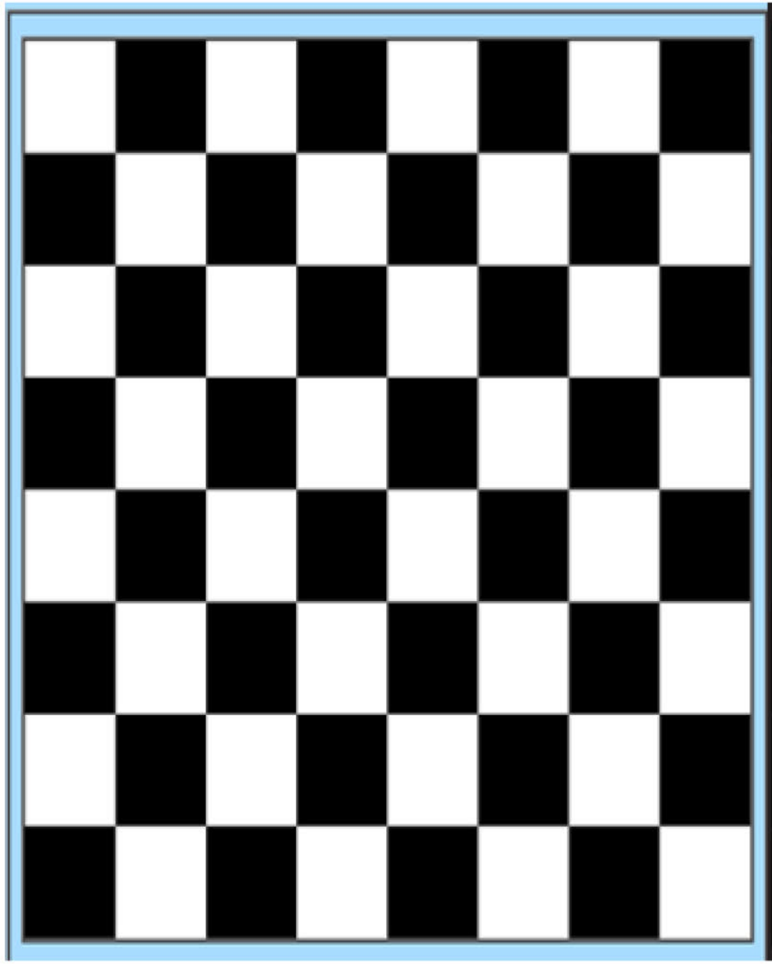
Farmer takes duck to left bank

Success!

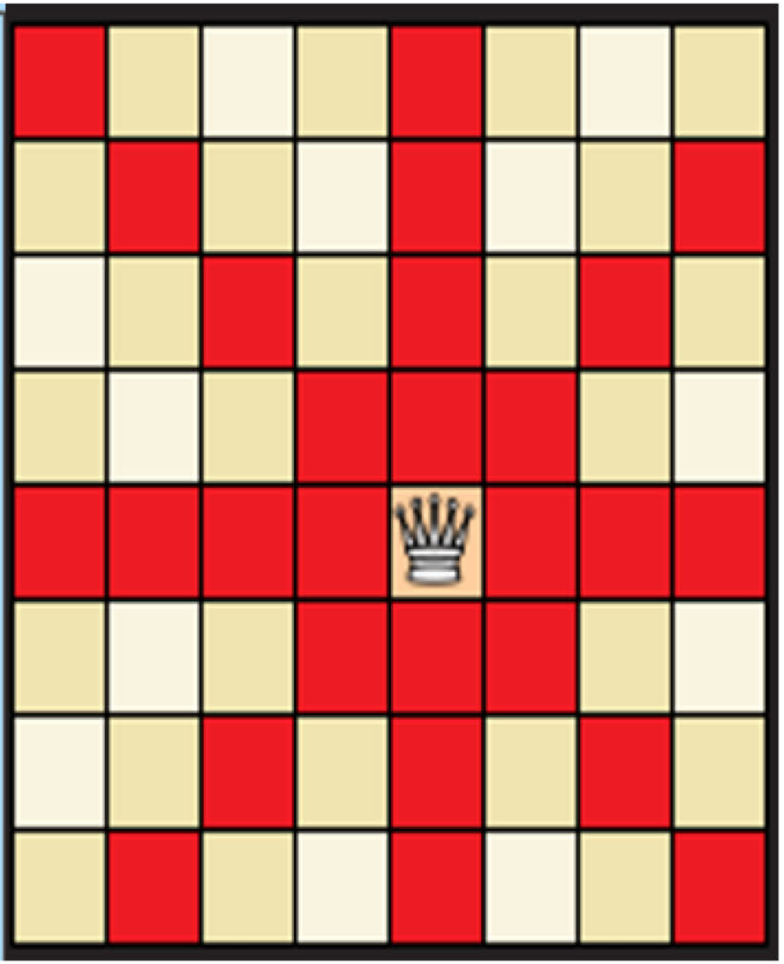
Eight Queens Problem

- The 8-queens problem can be defined as : Place 8 queens on an (8 by 8) chess board such that none of the queens attacks any of the others.
- Problem Formulation:
  - ***Initial state:*** *no queens on the board*
  - ***Successor Function:*** *Add a queen in any square*
  - ***Goal test:*** *8 queens on the board, none are attacked.*
  - ***Path cost:*** *1 per move*
- The initial state has 64 successors. Each of the states at the next level have 63 successors, and so on. We can restrict the search tree somewhat by considering only those successors where no queen is attacking each other. To do that we have to check the new queen against all existing queens on the board. The solutions are found at a depth of 8.

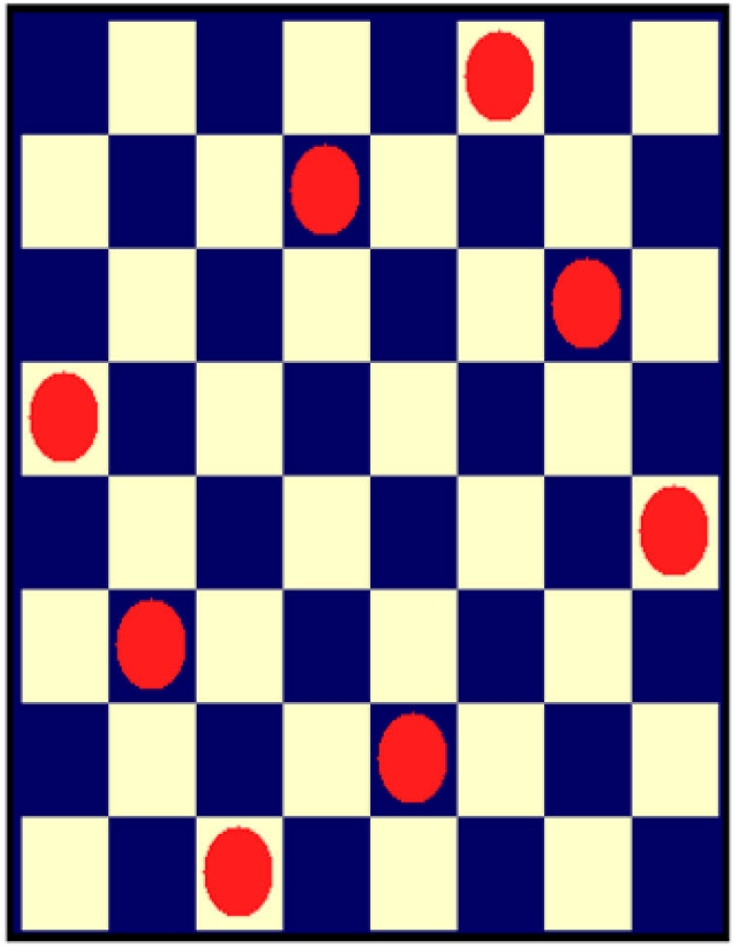
Eight Queens Problem



8*8 Chess Board

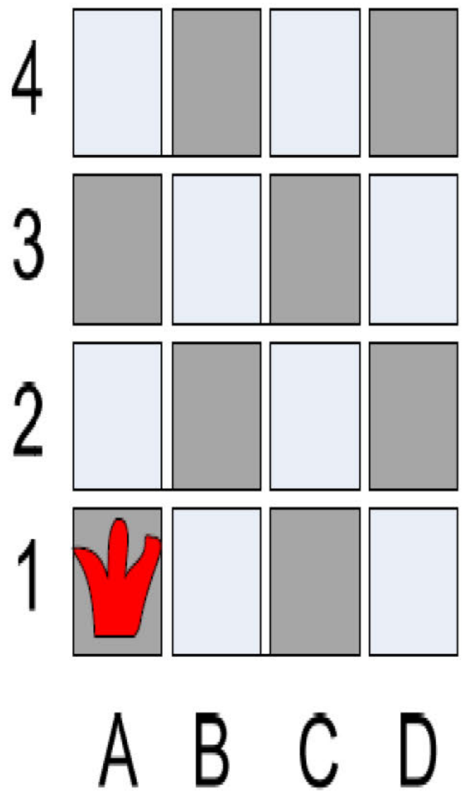


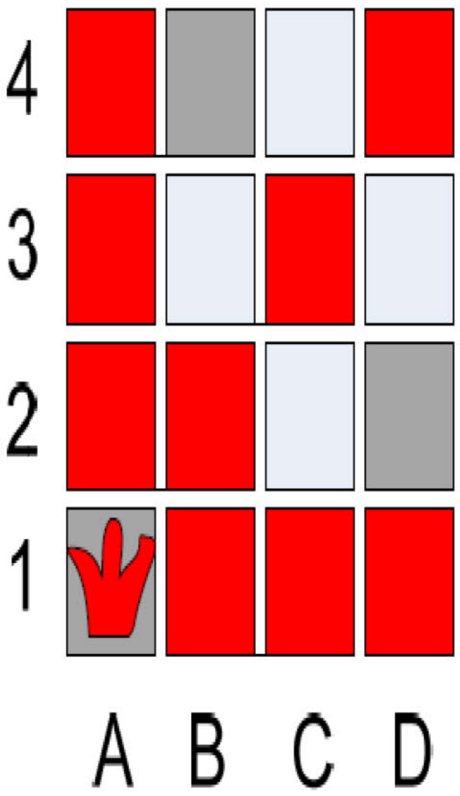
Possible Move of Queen

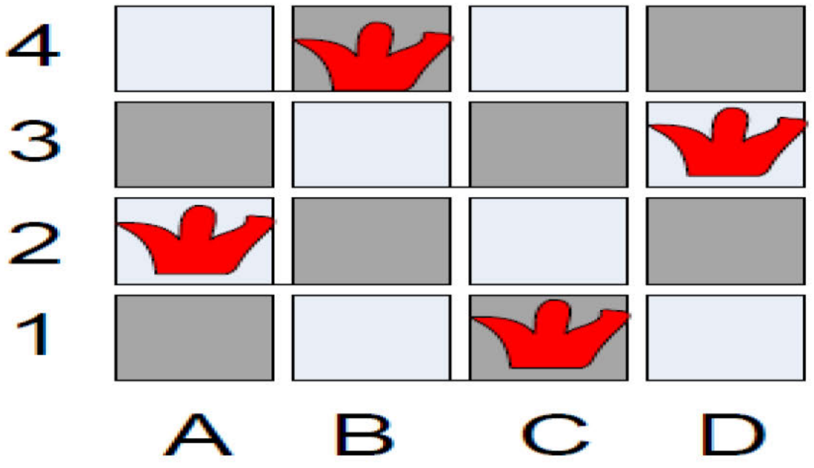


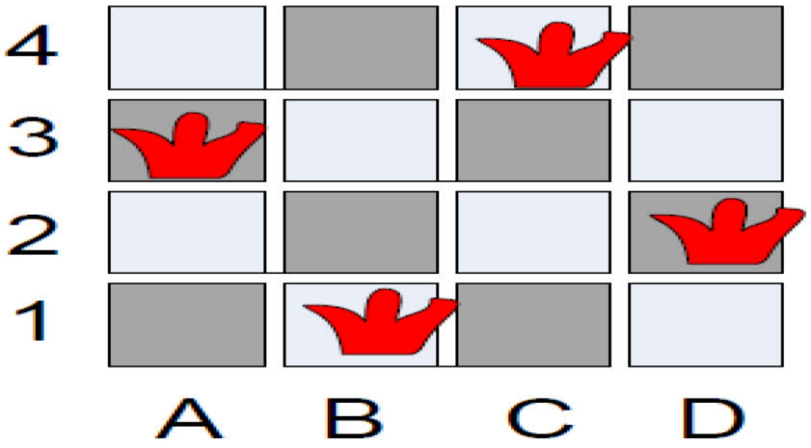
A Solution of 8*8 problem

*4*4 Queen problem: All Possible Solutions*









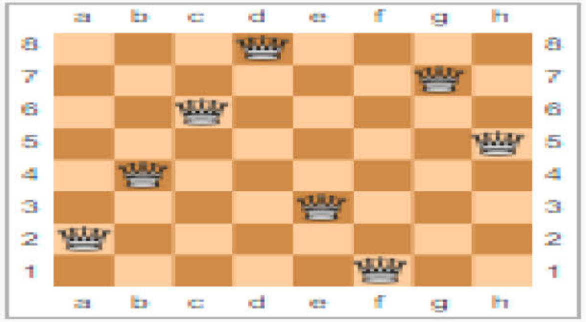
Eight Queens Problem

- **It is an iterative method. And For n>4 only. Let (*i*, *j*) be the square in column *i* and row *j* on the $n \times n$ chessboard , *k* an integer.**
  - If n is even and $n \neq 6k + 2$, then place queens at $(i, 2i)$ and $(n/2 + i, 2i - 1)$ for $i = 1,2,\ldots,n/2$.
  - If n is even and $n \neq 6k$, then place queens at $(i, 1 + (2i + n/2 - 3 \pmod{n}))$ and $(n + 1 - i, n - (2i + n/2 - 3 \pmod{n}))$ for $i = 1,2,\ldots,n/2$.
  - If n is odd, then use one of the patterns above for $(n - 1)$ and add a queen at (n, n).

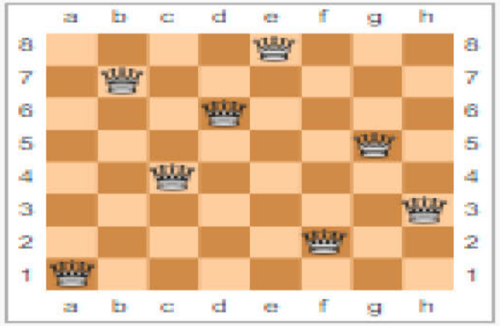
Eight Queens Problem

- **Another approach is**
  - If the remainder from dividing *N* by 6 is not 2 or 3 then the list is simply all even numbers followed by all odd numbers $\leq$ N
  - Otherwise, write separate lists of even and odd numbers (i.e. 2,4,6,8 - 1,3,5,7)
  - If the remainder is 2, swap 1 and 3 in odd list and move 5 to the end (i.e. **3**,**1**,7,**5**)
  - If the remainder is 3, move 2 to the end of even list and 1,3 to the end of odd list (i.e. 4,6,8,**2** - 5,7,**1**,**3**)
  - Append odd list to the even list and place queens in the rows given by these numbers, from left to right (i.e. a2, b4, c6, d8, e3, f1, g7, h5)
- **A few more examples follow.**
  - 14 queens (remainder 2): 2, 4, 6, 8, 10, 12, 14, 3, 1, 7, 9, 11, 13, 5.
  - 15 queens (remainder 3): 4, 6, 8, 10, 12, 14, 2, 5, 7, 9, 11, 13, 15, 1, 3.
  - 20 queens (remainder 2): 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 3, 1, 7, 9, 11, 13, 15, 17, 19, 5.

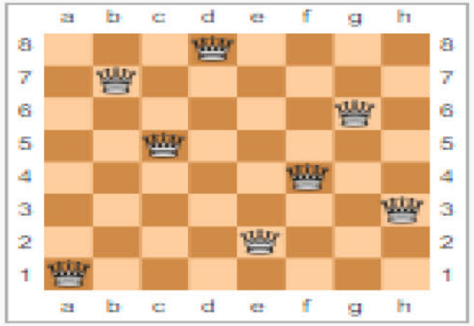
Eight Queens Problem



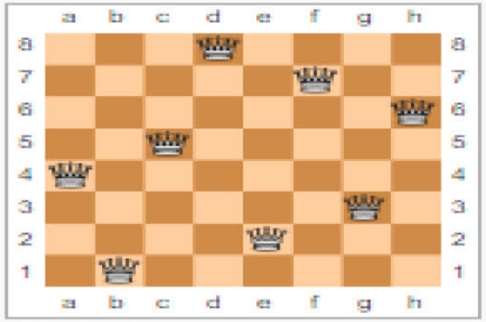
Solution 1



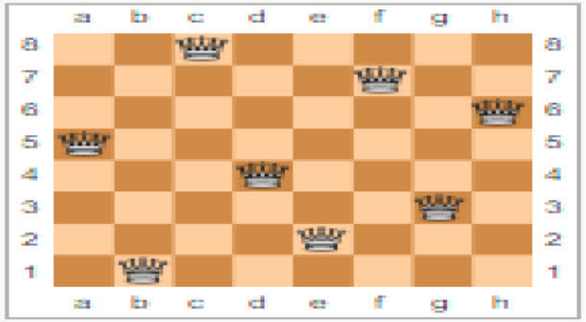
Solution 2



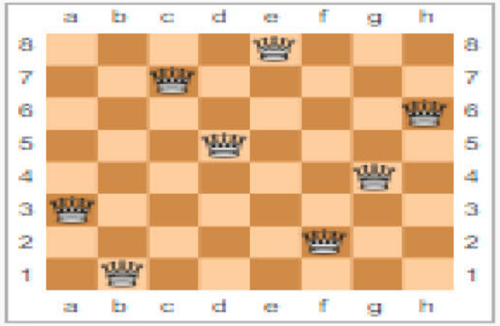
Solution 3



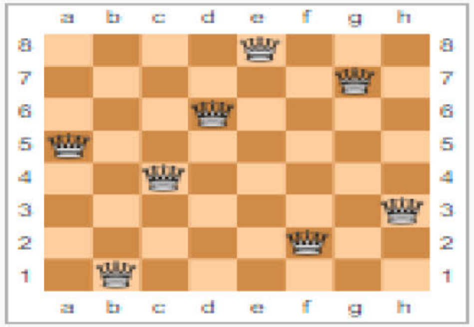
Solution 4



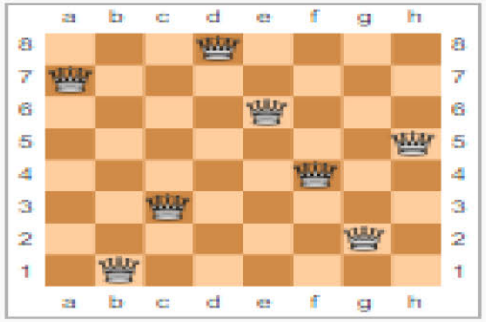
Solution 5



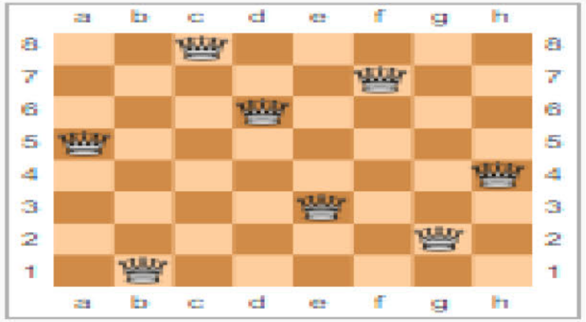
Solution 6



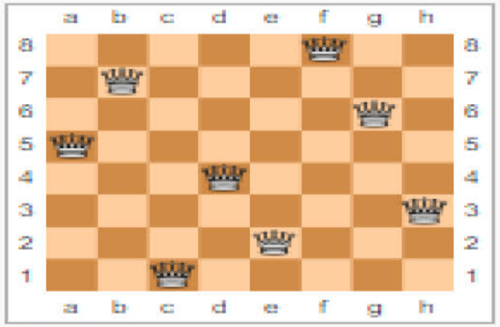
Solution 7



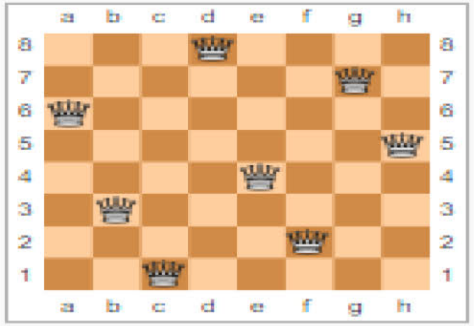
Solution 8



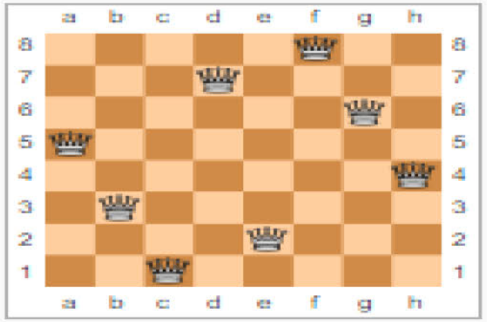
Solution 9



Solution 10



Solution 11



Solution 12

Eight Queens Problem

Board	Total Solutions	Unique Solutions
1 x 1	1	1
2 x 2	0	0
3 x 3	0	0
4 x 4	2	1
5 x 5	10	2
6 x 6	4	1
7 x 7	40	6
8 x 8	92	12
9 x 9	352	46
10 x 10	724	92
11 x 11	2,680	341
12 x 12	14,200	1,787
13 x 13	73,712	9,233
14 x 14	365,596	45,752
15 x 15	2,279,184	285,053
16 x 16	14,772,512	1,846,955
17 x 17	95,815,104	11,977,939
18 x 18	666,090,624	83,263,591
19 x 19	4,968,057,848	621,012,754
20 x 20	39,029,188,884	4,878,666,808
21 x 21	314,666,222,712	39,333,324,973
22 x 22	2,691,008,701,644	336,376,244,042
23 x 23	24,233,937,684,440	3,029,242,658,210
24 x 24	227,514,171,973,736	28,439,272,956,934
25 x 25	2,207,893,435,808,350	275,986,683,743,434
26 x 26	22,317,699,616,364,000	2,789,712,466,510,280

Constraints Satisfaction Problem

- A search procedure that operates in a space of constraints. Constraints are discovered and propagated as far as possible throughout the system. A guess about something is made and added as a new constraint.
- A constraint satisfaction problem is defined by a set of variables, $X_1, X_2, \ldots, X_n$, and a set of constraints, $C_1, C_2, \ldots, C_m$. Each variable X_i has a nonempty domain D_i of possible values.
- Each constraint C_i involves some subset of the variables and specifies the allowable combinations of values for that subset.
- ***Constraint propagation terminates for one of two reasons.***
    1. Contradiction detected i.e. no solution consistent with known constraints.
    2. Propagation has run off stream and there are no further changes that can be made on the basis of current knowledge.
- ***Example of such a problem are***
    1. ***Crypt-Arithmetic puzzles.***
    2. ***Map colouring:*** Given a map, colour three regions in blue, red and black, such that no two neighbouring regions have the same colour.

Example -1: Crypt-Arithmetic puzzle

- Solve the following puzzle by assigning numeral (0-9) in such a way that each letter is assigned unique digit which satisfy the following addition.

	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

- Initial Problem State : S = ? ; E = ? ;N = ? ; D = ? ; M = ? ;O = ? ; R = ? ;Y = ?

Carries :

$C_4 = ? ; C_3 = ? ; C_2 = ? ; C_1 = ?$

C_4	C_3	C_2	C_1	⟵	Carry
	S	E	N	D	
+	M	O	R	E	
M	O	N	E	Y	

Constraint equations:

$$Y = D + E \longrightarrow C_1$$
$$E = N + R + C_1 \longrightarrow C_2$$
$$N = E + O + C_2 \longrightarrow C_3$$
$$O = S + M + C_3 \longrightarrow C_4$$
$$M = C_4$$

Example -1 : Crypt-Arithmetic puzzle

- We can easily see that M has to be non zero digit, so the value of $C4 = 1$
  1. $M = C4 \Rightarrow \mathbf{M = 1}$
  2. $O = S + M + C3 \rightarrow C4$
     - For $C4 = 1$, $S + M + C3 > 9 \Rightarrow S + 1 + C3 > 9 \Rightarrow S + C3 > 8$.
     - If $C3 = 0$, then $S = 9$ else if $C3 = 1$, then $S = 8$ or 9.
     - We see that for $S = 9$ and $C3 = 0$ or 1
     - It can be easily seen that $C3 = 1$ is not possible as $O = S + M + C3 \Rightarrow O = 11 \Rightarrow O$ has to be assigned digit1 but 1 is already assigned to M, so not possible.
     - Therefore, only choice for $C3 = 0$, and thus $O = 10$. This implies that O is assigned 0 (zero) digit.
     - **Therefore, $\mathbf{M = 1, O = 0}$**
  3. Since $C3 = 0$; $N = E + O + C2$ produces no carry.
  - As $O = 0$, $N = E + C2$. Since $N \neq E$, therefore, $C2 = 1$. **Hence $\mathbf{N = E + 1}$**

Example -1 : Crypt-Arithmetic puzzle

- Now E can take value from 2 to 8 {0,1,9 already assigned so far }
- If E = 2, then N = 3.
  - Since C2 = 1, from E = N + R + C1 , we get 12 = N + R + C1
    - ✓ If C1 = 0 then R = 9, which is not possible as we are on the path with S = 9
    - ✓ If C1 = 1 then R = 8, then
      - ➢ From Y = D + E , we get 10 + Y= D + 2 .
      - ➢ For no value of D, we can get Y.
    - ✓ Try similarly for E = 3, 4. We fail in each case.
- If E = 5, then N = 6
  - Since C2 = 1, from E = N + R + C1 , we get 15 = N + R + C1 ,
  - If C1 = 0 then R = 9, which is not possible as we are on the path with S = 9.
  - If C1 = 1 then R = 8, then
    - ✓ From Y = D + E , we get 10 + Y= D + 5 i.e., 5 + Y = D.
    - ✓ If Y = 2 then D = 7. These values are possible.

S = 9 ; E = 5 ; N = 6 ; D = 7 ; M = 1 ; O = 0 ; R = 8 ;Y = 2

Example -2: Crypt-Arithmetic puzzle

C4	C3	C2	C1		← Carries
	B	A	S	E	
+	B	A	L	L	
G	A	M	E	S	

Constraints equations are:

$E + L = S \rightarrow C1$

$S + L + C1 = E \rightarrow C2$

$2A + C2 = M \rightarrow C3$

$2B + C3 = A \rightarrow C4$

$G = C4$

Initial Problem State : G = ?; A = ?;M = ?; E = ?; S = ?; B = ?; L = ?

1. $G = C_4 \Rightarrow \mathbf{G = 1}$

2. $2B + C_3 = A \quad \rightarrow C_4$

   2.1 Since $C_4 = 1$, therefore, $2B + C_3 > 9 \Rightarrow B$ can take values from 5 to 9.

   2.2 Try the following steps for each value of B from 5 to 9 till we get a possible value of B.

- If $B = 5$
  - if $C_3 = 0 \Rightarrow A = 0 \Rightarrow M = 0$ for $C_2 = 0$ or $M = 1$ for $C_2 = 1$ $\times$
  - if $C_3 = 1 \Rightarrow A = 1$ $\times$ (as $G = 1$ already)
- For $B = 6$ we get similar contradiction while generating the search tree.
- If $\boxed{B = 7}$, then for $C_3 = 0$, we get $\boxed{A = 4}$ $\Rightarrow M = 8$ if $C_2 = 0$ that leads to contradiction, so this path is pruned. If $C_2 = 1$, then $\boxed{M = 9}$.

3. Let us solve $S + L + C_1 = E$ and $E + L = S$
   - Using both equations, we get $2L + C_1 = 0 \Rightarrow \boxed{L = 5}$ and $C_1 = 0$
   - Using $L = 5$, we get $S + 5 = E$ that should generate carry $C_2 = 1$ as shown above
   - So $S+5 > 9 \Rightarrow$ Possible values for E are {2, 3, 6, 8} (with carry bit $C_2 = 1$)
   - If $E = 2$ then $S + 5 = 12 \Rightarrow S = 7$ $\times$ (as $B = 7$ already)
   - If $E = 3$ then $S + 5 = 13 \Rightarrow S = 8$.
   - Therefore $\boxed{E = 3}$ and $\boxed{S = 8}$ are fixed up.

4. Hence we get the final solution as given below and on backtracking, we may find more solutions. In this case we get only one solution.

   G = 1; A = 4; M = 9; E = 3; S = 8; B = 7; L = 5

Thank You
???

- ***References:***

- *Stuart Russell, Peter Norvig - Artificial Intelligence_ A Modern Approach-Prentice Hall (2010)*
- *Kevin Knight, Elaine Rich, B. Nair - Artificial Intelligence-Tata McGraw-Hill Education Pvt. Ltd. (2010)*
- *James A. Freeman, David M. Skapura - Neural networks, algorithms, applications, and programming techniques-Addison-Wesley Pub (Sd) (1991)*