

1. Problem Solving using Search

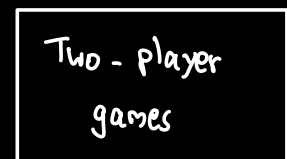
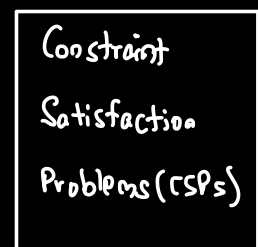
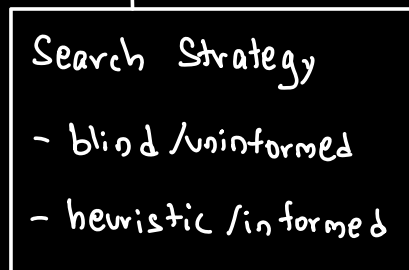
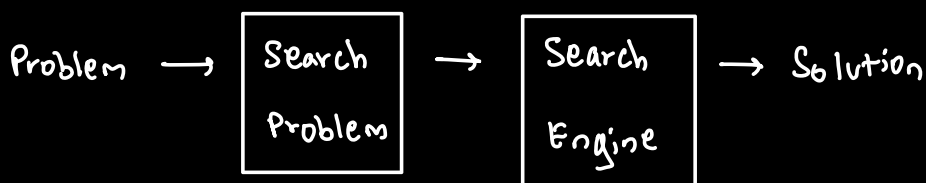
Knowledge Representation and reasoning

2. - Symbolic (logic)

3. - Numeric (Probability)

4. Machine Learning.

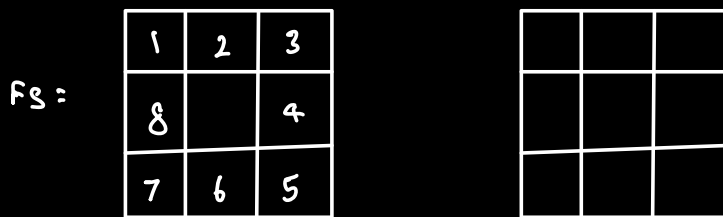
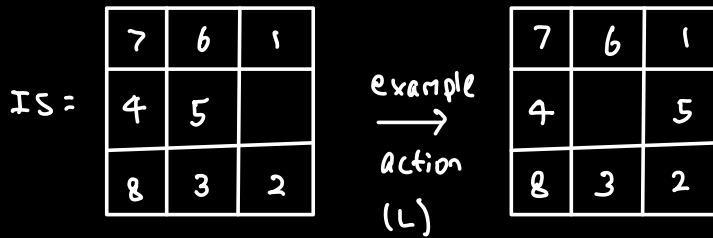
PROBLEM SOLVING USING SEARCH:



- 1. Initial State (IS)
 - 2. Final State (FS)
 - 3. Actions

Enough for blind strategy
4. Heuristic

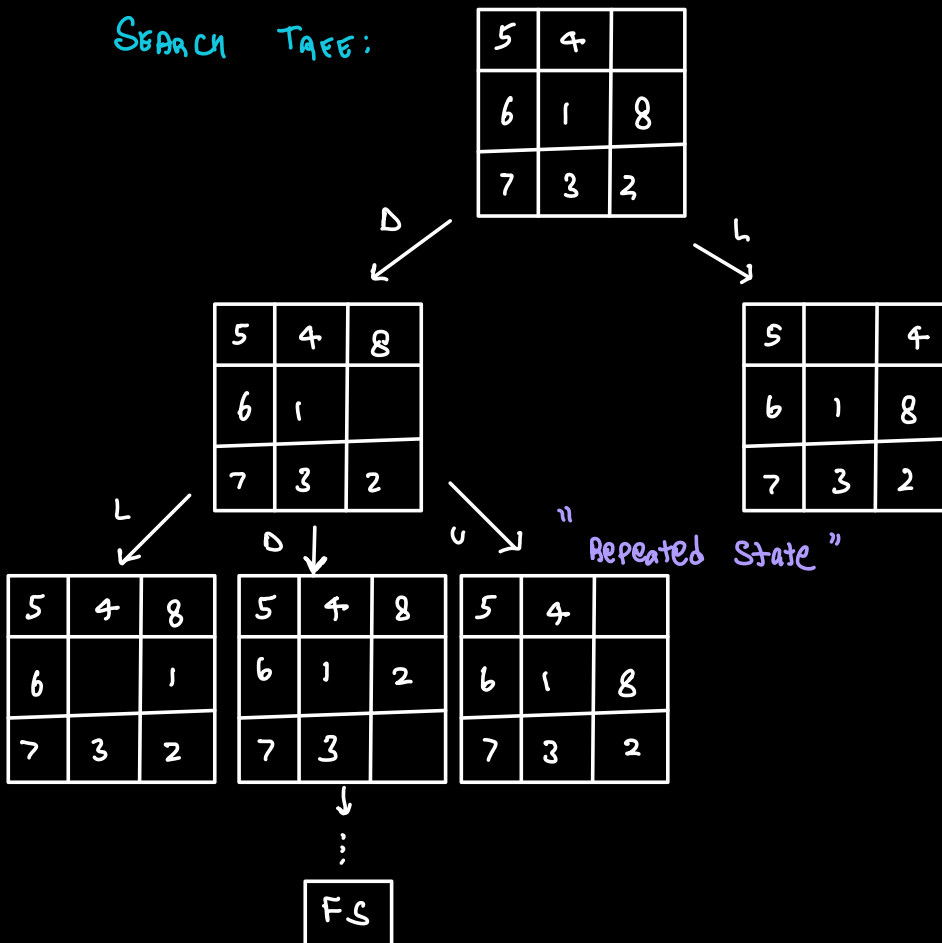
8- PUZZLE:



Actions: U, D, L, R.

We can move the blank up, down, left or right.

SEARCH TREE:

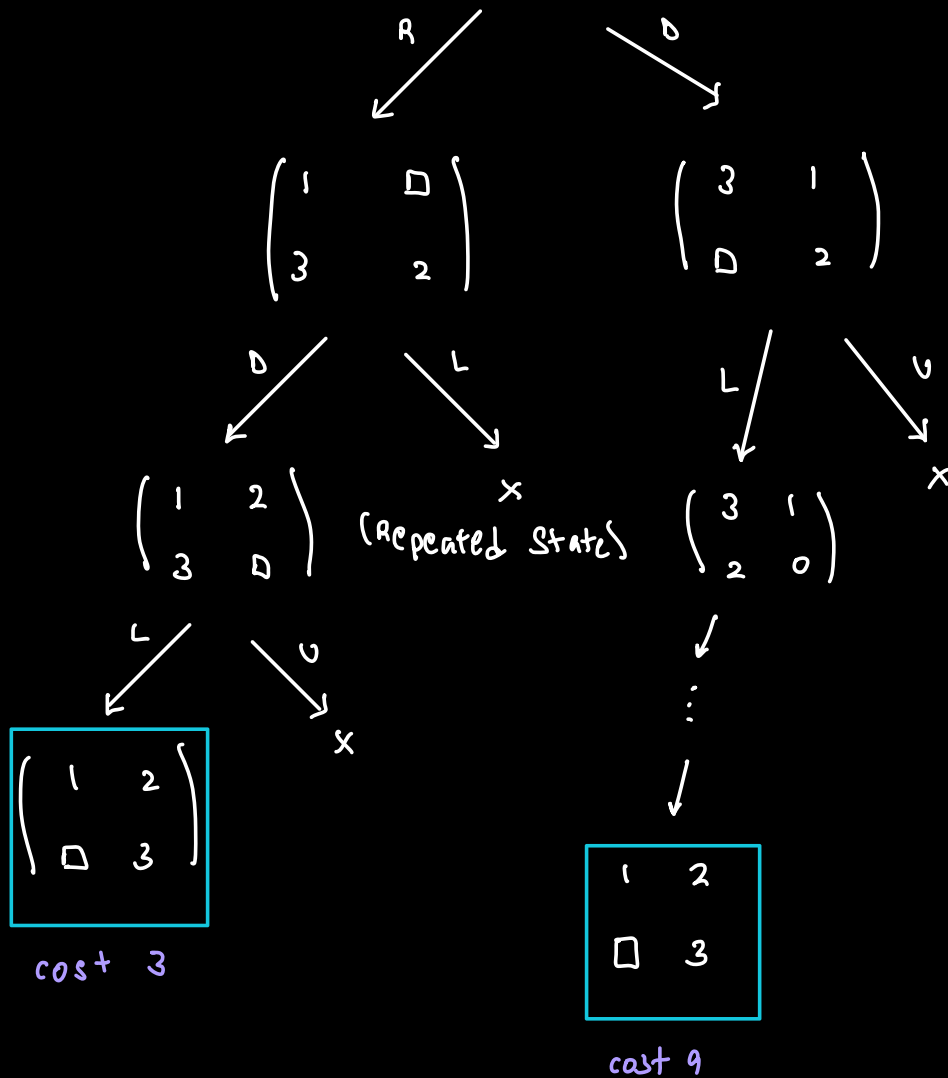


Goal: Solve without creating a search tree.

Cost: length of the path from IS to FS.

$$FS \begin{pmatrix} 1 & 2 \\ 0 & 3 \end{pmatrix}$$

$$IS \begin{pmatrix} 0 & 1 \\ 3 & 2 \end{pmatrix}$$

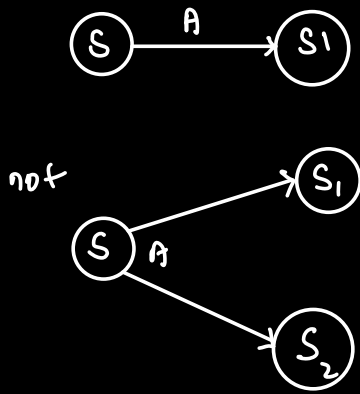


OPTIMIZE:

- (1) Cost of a solution.
- (2) Cost of finding a solution.

ASSUMPTIONS:

- (1) Actions are deterministic



- (2) Know the initial state.

PLANNING

→ Conditional Plans

→ Conformant Plans.

RIVER / BOAT:

IS =

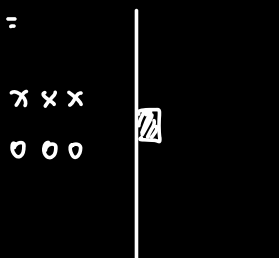


boat

x → good guys

o → bad guys

FS =

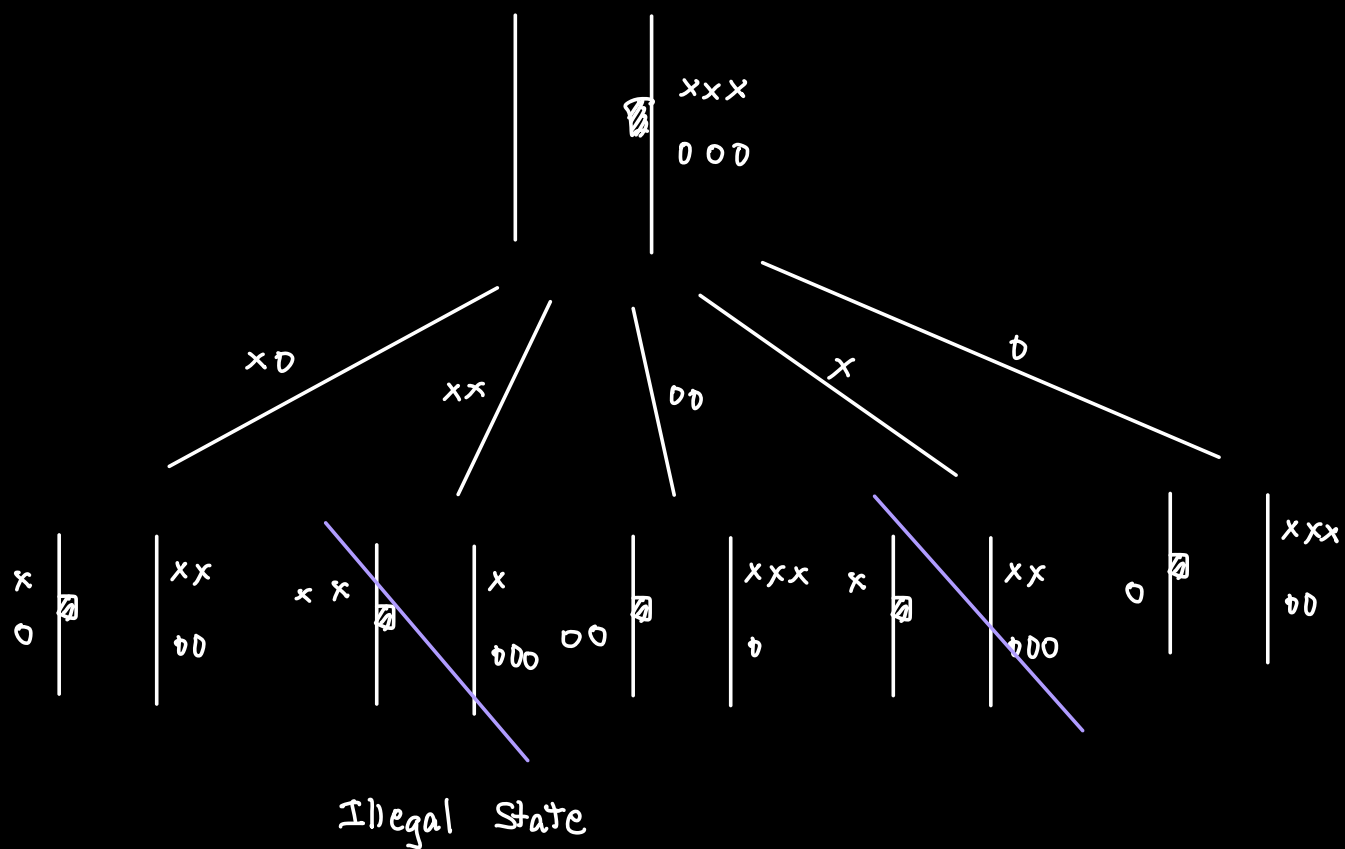


Rules:

- * Boat can take at most 2
- * Cannot have more o's on one side than x's.
- * We can have all o's on one side.
- * Boat cannot travel empty.

Actions:

x, o, xx, oo, xo.



COMMUNICATING WITH THE SEARCH ENGINE:

- ### 1. Initial State

- ## 2. Final State

- ### 3. Actions

(Legal states and actions)

(Successor function)

State

$$(a \quad b \quad c)$$

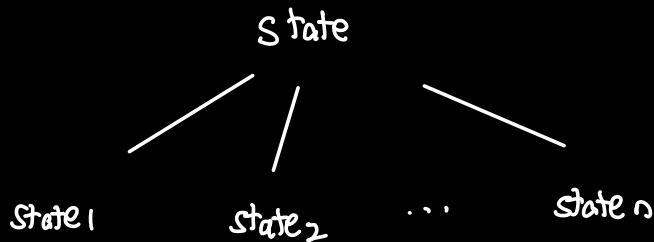
a : number of x at boat location

b: number of 0 at boat location

c : boat location → t right
 └→ NIL left

$$I_S \quad (3 \quad 3 \quad t)$$

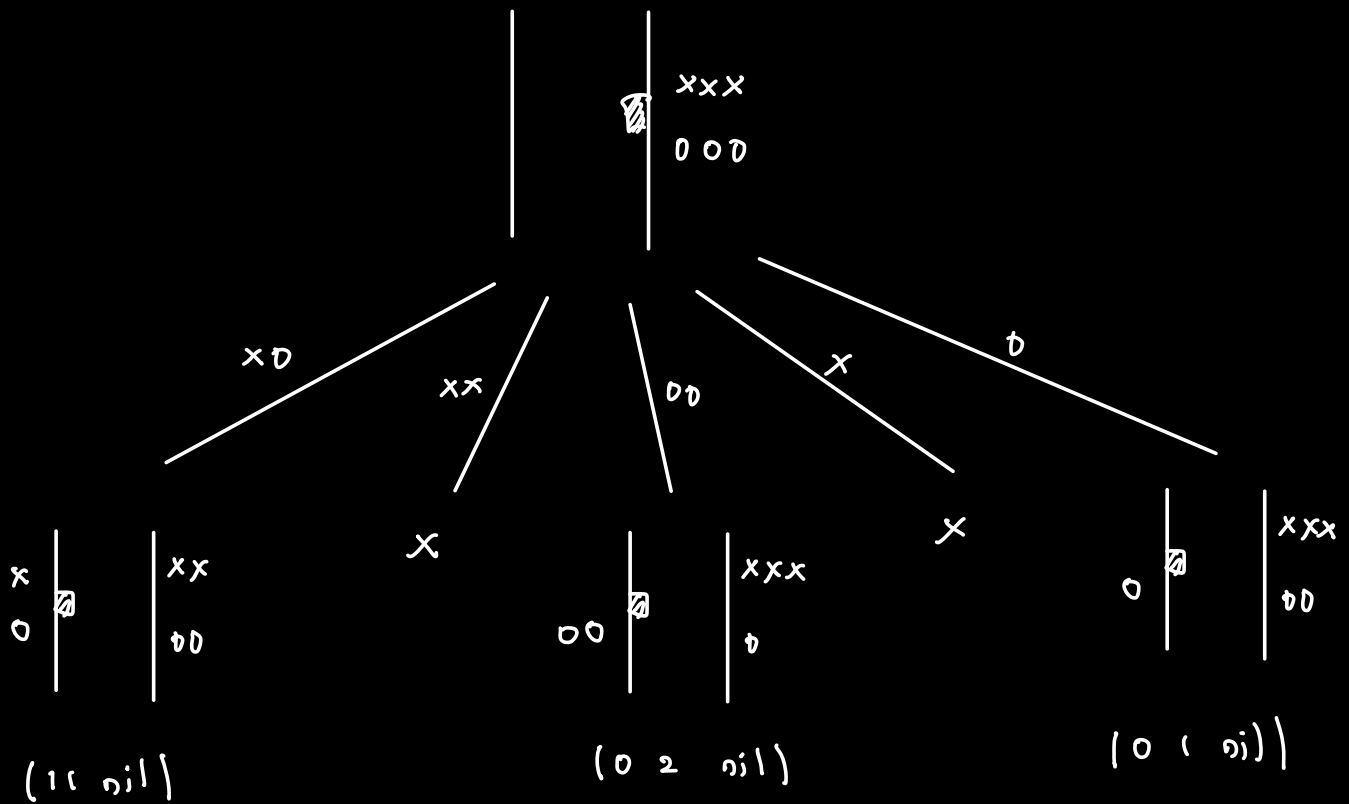
$F_S \quad (3 \quad 3 \quad \text{NIL})$



Successor
function.



As returns list of
legal state.



> (succ-fn '(3 3 t))

> ((1 1 nil) (0 2 nil) (0 1 nil))

Complexity:

- * Number of branching factors.

- * Search Space - number of distinct states.

- * Depth at which solution exists.

CONSTRAINT SATISFACTION PROBLEMS (CSPs):

Q							
				Q			
	Q						
					Q		
		Q					
						Q	
			Q				
							Q

No two queens

1. in the same row
2. in the same column
3. in the same diagonal

- Initial State

Empty Board

- Final State

8 Queens. Valid Board.

- Actions.

Add a queen.

1. Have a function that says if a state is final state.
2. Finite depth. All solutions at same depth.