18CSC305J ARTIFICIAL INTELLIGENCE

1

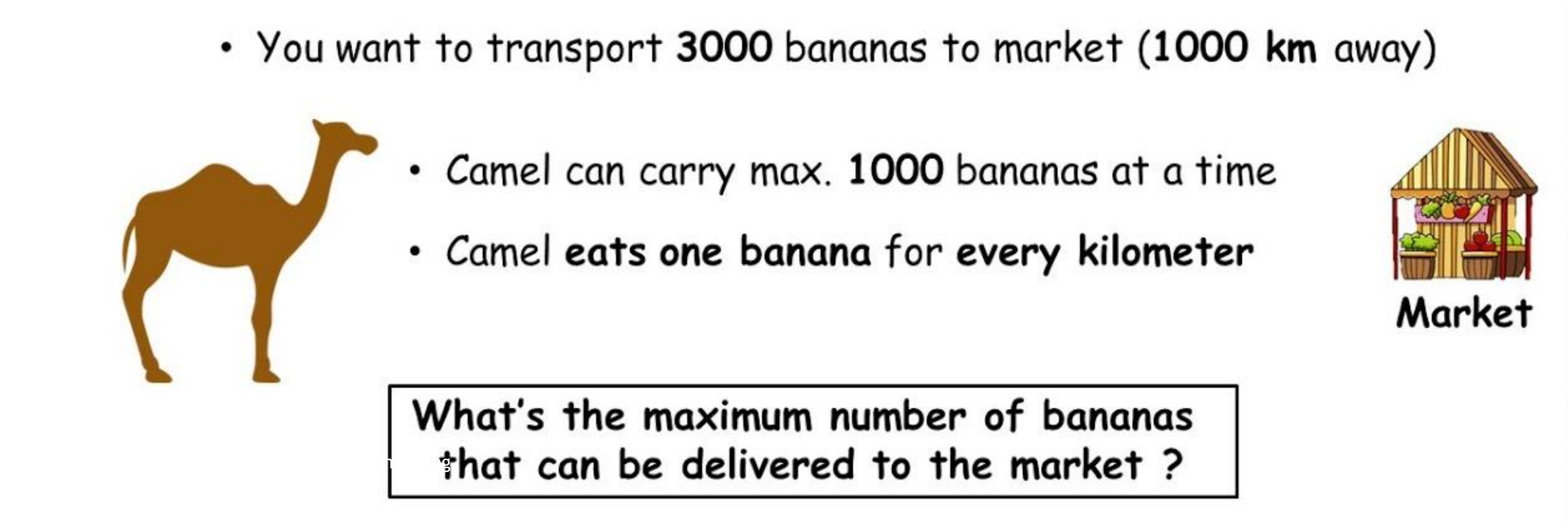
2018 Regulation
Third year- 6^{th} sem

School of Computing
SRM Institute of Science & Technology
Kattankulathur Campus

-Vaishnavi Moorthy, AP/School of Computing

Lab Exercise-1:Toy Problem (Any)

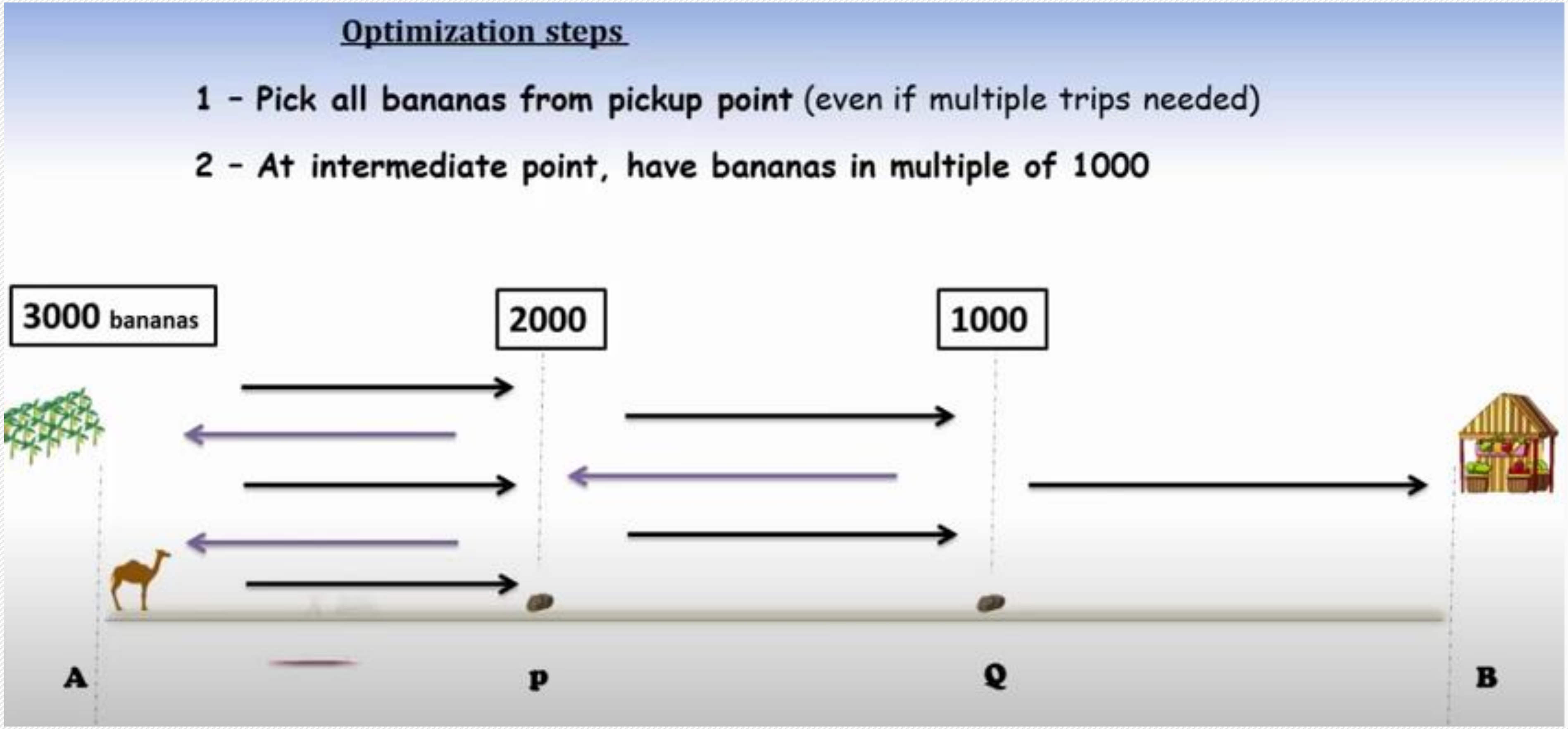
Camel and Banana Puzzle



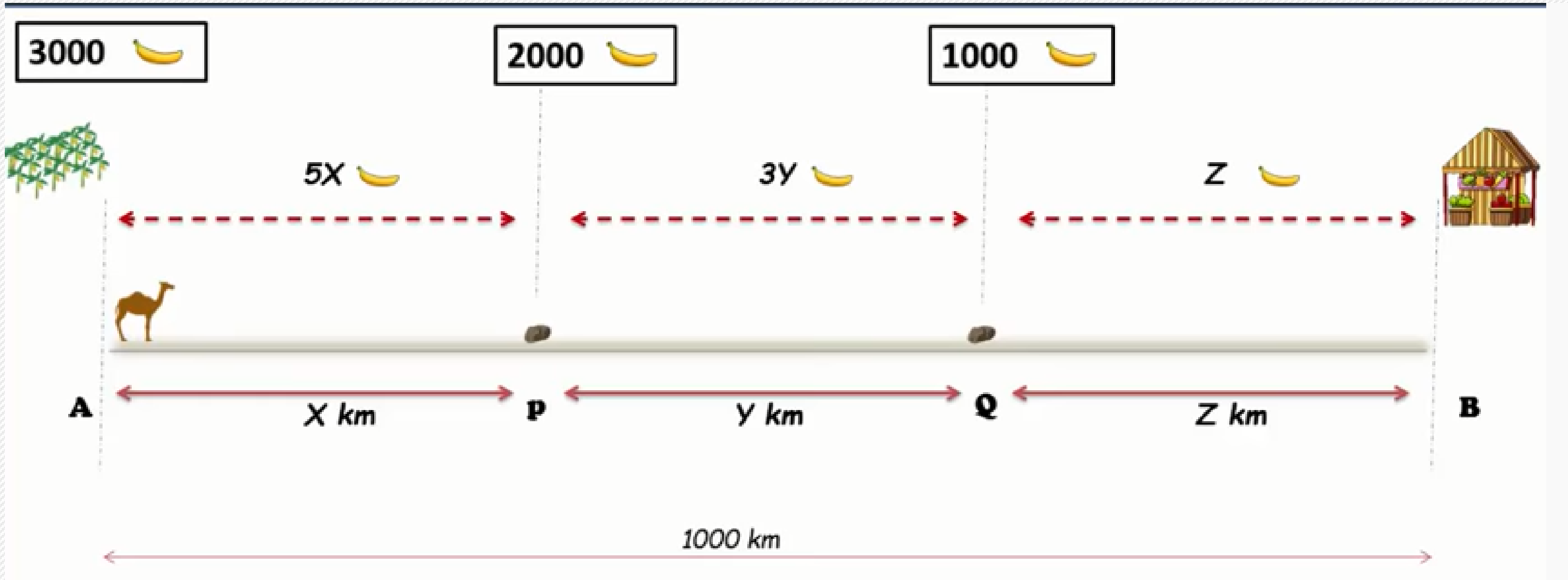
Solution & Inferences

- We have a total of 3000 bananas.
- The destination is 1000KMs
- Only 1 mode of transport.
- Camel can carry a maximum of 1000 banana at a time.
- Camel eats a banana every km it travels.
- One complete trip means output zero
  - Solution: Intermediate points (2 in this case to have in multiples of 1000)

Optimization steps



Optimization steps



$3000 - 5x = 2000$

$X = 200$ kms

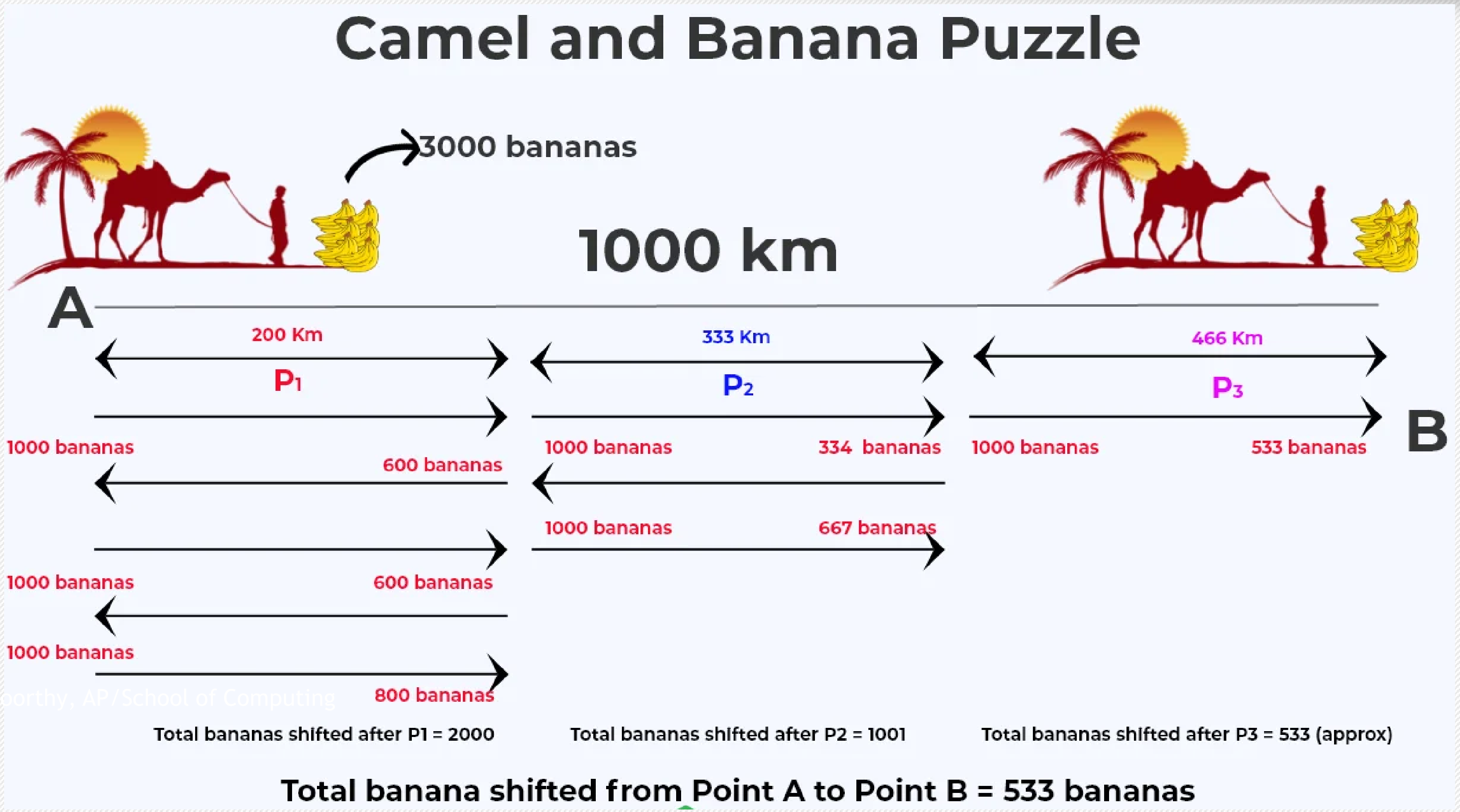
$2000 - 3y = 1000$

$Y = 333.33 = 333$ kms

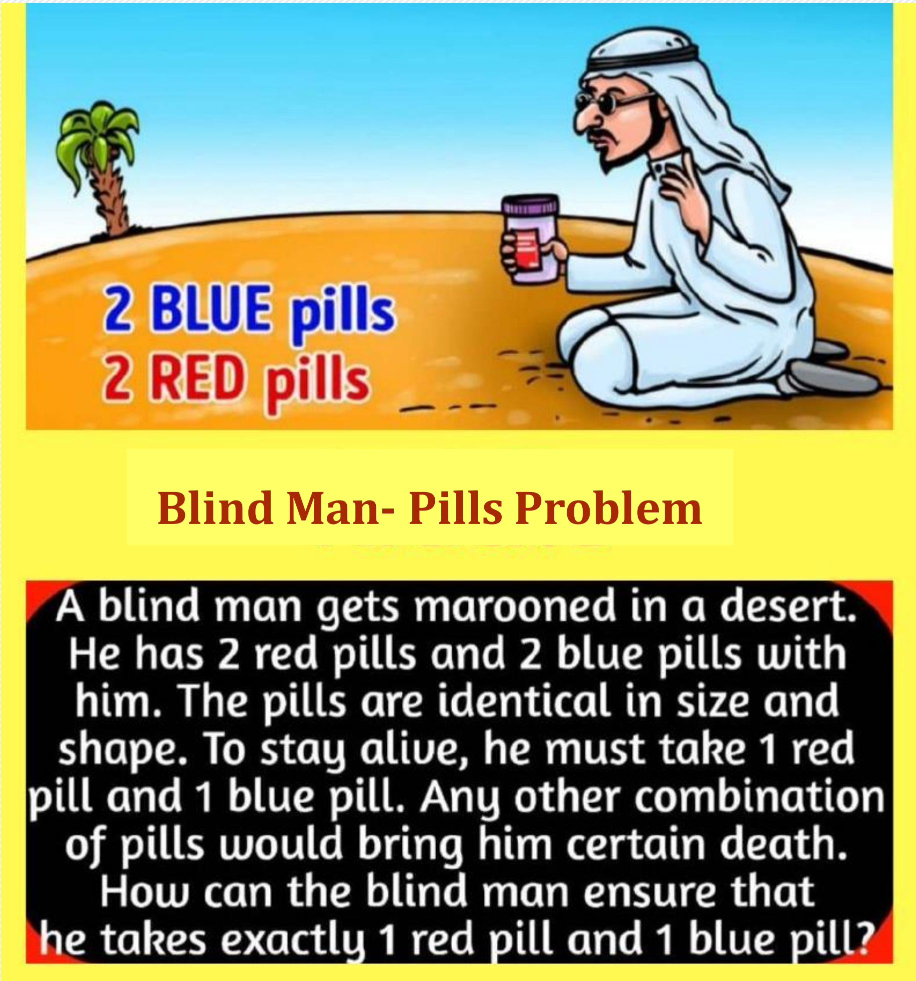
$Z = 1000 – 200 – 333$

$Z = 467$ kms

Optimization steps

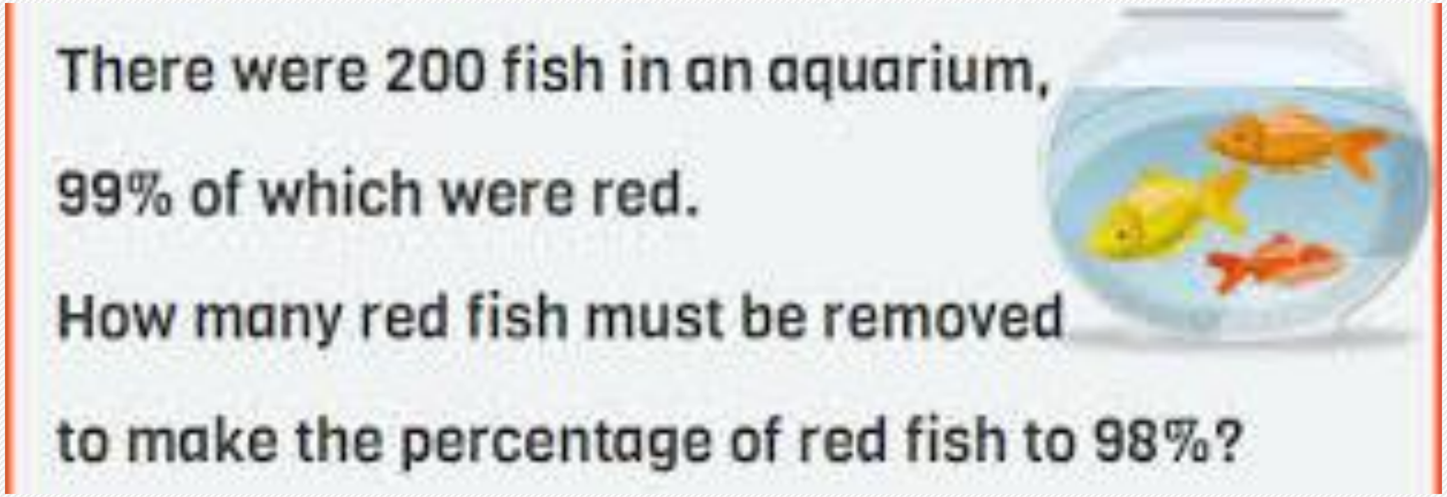


Other Examples



Blind Man- Pills Problem

Red fish puzzle



Other Examples

River Crossing

Two boys wish to cross a river. The only way to get to the other side is by boat, but that boat can only take one boy at a time. The boat cannot return on its own, there are no ropes or similar tricks, yet both boys manage to cross using the boat. How?

Prisoner hat problem

Three prisoners are told that at midnight, in the dark, each will be fitted with a red or blue hat according to a fair coin flip. The lights will then briefly be turned on, enabling each prisoner to see every other prisoner's hat color. Once the lights are on, however, the prisoners will have no opportunity to signal to one another or to communicate in any way.

Each prisoner will then be taken aside and given the option of trying to guess whether his or her own hat is red or blue, but he or she may choose to pass. All three prisoners will be freed if (1) at least one prisoner chooses to guess his or her hat color, and (2) every prisoner who chooses to guess guesses correctly.

The prisoners have a chance to devise a strategy before the game begins. Can they achieve a winning probability greater than 50%?

List of Toy problems

- Minimum cut Puzzle
- torch and bridge puzzle problem
- 10 coins puzzle
- Hourglasses puzzle
- Vacuum world
- N-puzzle problem
- cost to buy all books
- Maximum Subarray Problem
- buy sell stock puzzle
- total distance travelled by bee
- 3 Priests 3 devils puzzle problem
- Camel and Bananas Problem
- The wolf-goat-cabbage problem
- SuperMarket Queue
- Trapping rain water
- Two Burner Dishes Problem

Toy Problem- 1

Vacuum Cleaner World -Problem Formulation

- State - agent is in one of two locations, each of which might or might not contain dirt. Thus there are $2 \times 2^2 = 8$ possible world states. A larger environment with n locations has $n \cdot 2^n$ states

- Initial State

– Any one of 8 states

- Actions

— In this simple environment, each state has just three actions: *Left* , *Right* ,*Suck*. Larger environments might also include *Up* , *Down*

Toy Problem- 1

Vacuum Cleaner World -Problem Formulation

- **Transition model**: The actions have their expected effects, except that moving *Left* in the leftmost square, moving *Right* in the rightmost square, and *Suck*ing in a clean square have no effect. The complete state space is shown in the figure .

- State Space for the Vacuum World

Labels on Arcs denote L: Left, R: Right, S: Suck

- Goal Test

– This checks whether all the squares are clean

- Path Cost

– Number of steps (each step costs a value of 1)

Toy Problem- 2

The 8-Puzzle (Sliding Block Puzzle)

The **8-puzzle**, an instance of which is shown below, consists of a 3×3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space. The object is to reach a specified goal state, such as the one shown on the right of the figure.

1	2	3
7	8	4
6		5

Initial State

1	2	3
8		4
7	6	5

Goal State

Toy Problem- 2
The 8-Puzzle (Sliding Block Puzzle)

States: A state description specifies the location of each of the eight tiles and the blank in one of the nine squares.

Initial state: Any state can be designated as the initial state.

Successor function: This generates the legal states that result from trying the four actions (blank moves Left, Right, Up, or Down).

Goal test: This checks whether the state matches the goal configuration (Other goal configurations are possible.)

Path cost: Each step costs 1, so the path cost is the number of steps in the path.

Toy Problem- 2

The 8-Puzzle (Sliding Block Puzzle) - Solution

- **h_f= +1** for every correct position
- Solution of this problem is "movement of tiles" in order to reach goal state.
- The transition function or legal move is any one tile movement by one space in any direction.

1	2	3
7	8	4
6		5

1	2	3
8		4
7	6	5

Goal State

1	2	3
7	8	4
	6	5

Move 1
h_f=6

1	2	3
7	8	4
6	5	

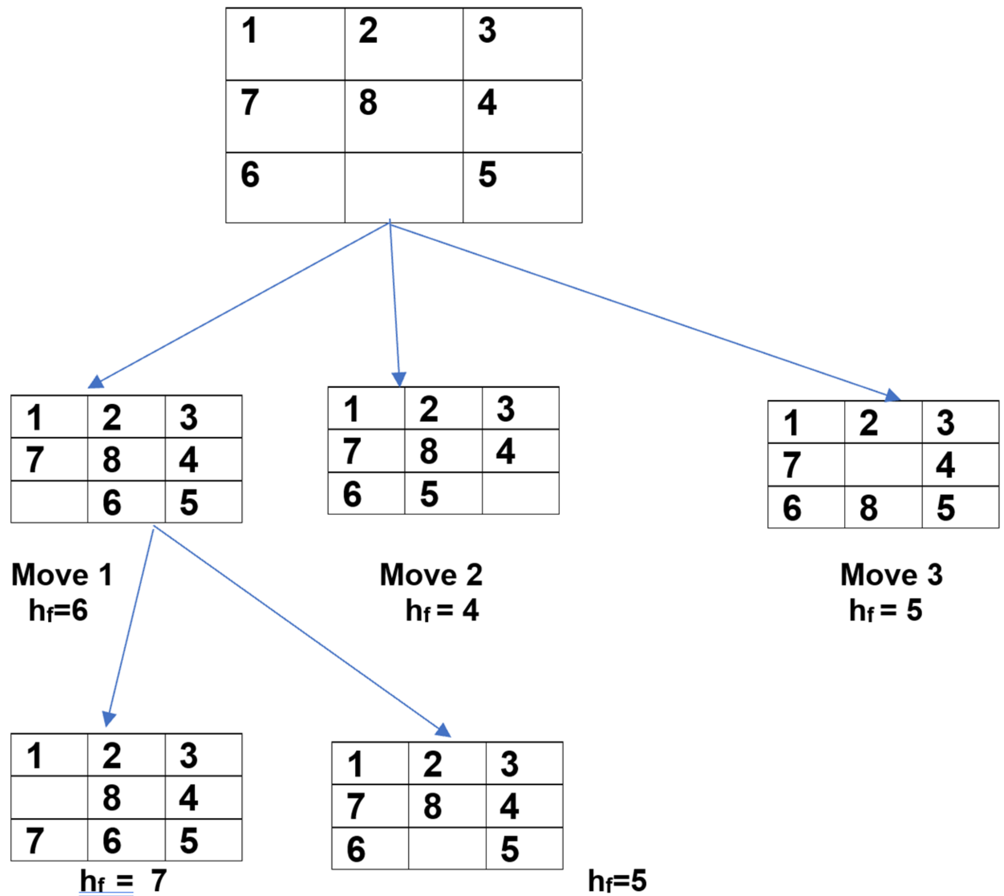
Move 2
h_f = 4

1	2	3
7		4
6	8	5

Move 3
h_f = 5

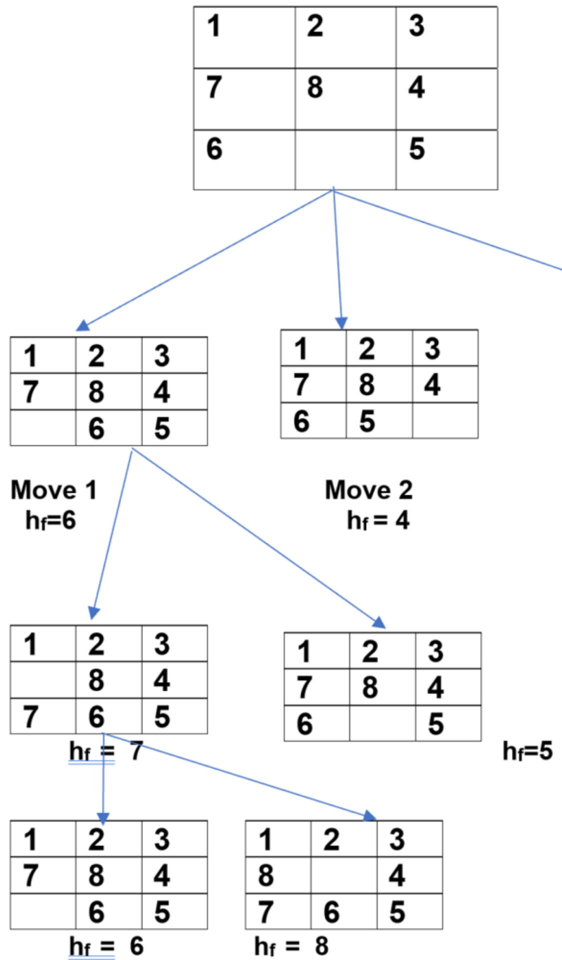
Toy Problem- 2

The 8-Puzzle (Sliding Block Puzzle) - Solution



Toy Problem- 2

The 8-Puzzle (Sliding Block Puzzle) - Solution



Toy Problem- 3

Water – Jug Problem

A Water Jug Problem: You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. How can you get exactly 2 gallons of water in the 4-gallon jug?

Toy Problem- 3

Water – Jug Problem

Solution:

The state space for this problem can be described as the set of ordered pairs of integers **(x,y)**

Where,

X represents the quantity of water in the 4-gallon jug **X= 0,1,2,3,4**
Y represents the quantity of water in 3-gallon jug **Y=0,1,2,3**

Note

$0 \leq X \leq 4$, and $0 \leq Y \leq 3$

Start State: (0,0)
Goal State: **(2, n)** for any n. Attempting to end up in a goal state.(since the problem doesn‘t specify the quantity of water in 3-gallon jug)

Toy Problem- 3

Water – Jug Problem

Generate production rules for the water jug problem

Production Rules:

1. (x,y) -> (4,y) Fill x
2. (x,y) -> (x,3) Fill y
3. (x,y) -> (x-d, y) Pour water out from X
4. (x,y) -> (x,y-d) Pour water from y
5. (x,y) -> (0,y) Empty x
6. (x,y) -> (x,0) Empty y
7. (x,y) -> (4,y-(4-x)) Pour water from y into x until x is full
8. (x,y) -> (x – (3-y), 3) Pour water from x into y until y is full.
9. (x,y) -> (x+y, 0) Pour all water from y to x
10. (x,y) -> (0, x+y) Pour all water from x to y
11. (0,2) -> (2,0) Pour 2 Gallon of water from y to x
12. (2, y) -> (0,y) Pour 2 Gallon of water from x to ground.

Toy Problem- 3
Water – Jug Problem

First solution

	4-g jug	3-g jug
Initial	0	0
R2	0	3
R9	3	0
R2	3	3
R7	4	2
R5	0	2
R9	2	0

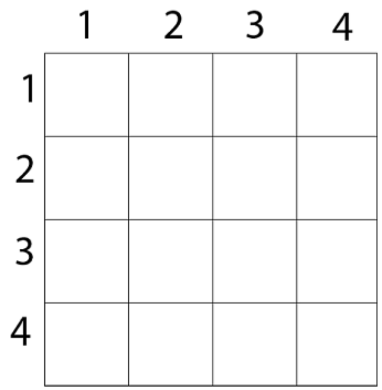
Goal State

1. (x,y) -> (4,y) Fill x
2. (x,y) -> (x,3) Fill y
3. (x,y) -> (x-d, y) Pour water out from X
4. (x,y) -> (x,y-d) Pour water from y
5. (x,y) -> (0,y) Empty x
6. (x,y) -> (x,0) Empty y
7. (x,y) -> (4,y-(4-x)) Pour water from y into x until x is full
8. (x,y) -> (x – (3-y), 3) Pour water from x into y until y is full.
9. (x,y) -> (x+y, 0) Pour all water from y to x
10. (x,y) -> (0, x+y) Pour all water from x to y
11. (0,2) -> (2,0) Pour 2 Gallon of water from y to x
12. (2, y) -> (0,y) Pour 2 Gallon of water from x to ground.

Toy Problem- 4(a)
4-queens problem

The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other.

Given a 4 x 4 chessboard and number the rows and column of the chessboard 1 through 4.



4x4 chessboard

Since, we have to place 4 queens such as q_1 q_2 q_3 and q_4 on the chessboard, such that no two queens attack each other. In such a conditional each queen must be placed on a different row, i.e., we put queen "i" on row "i."

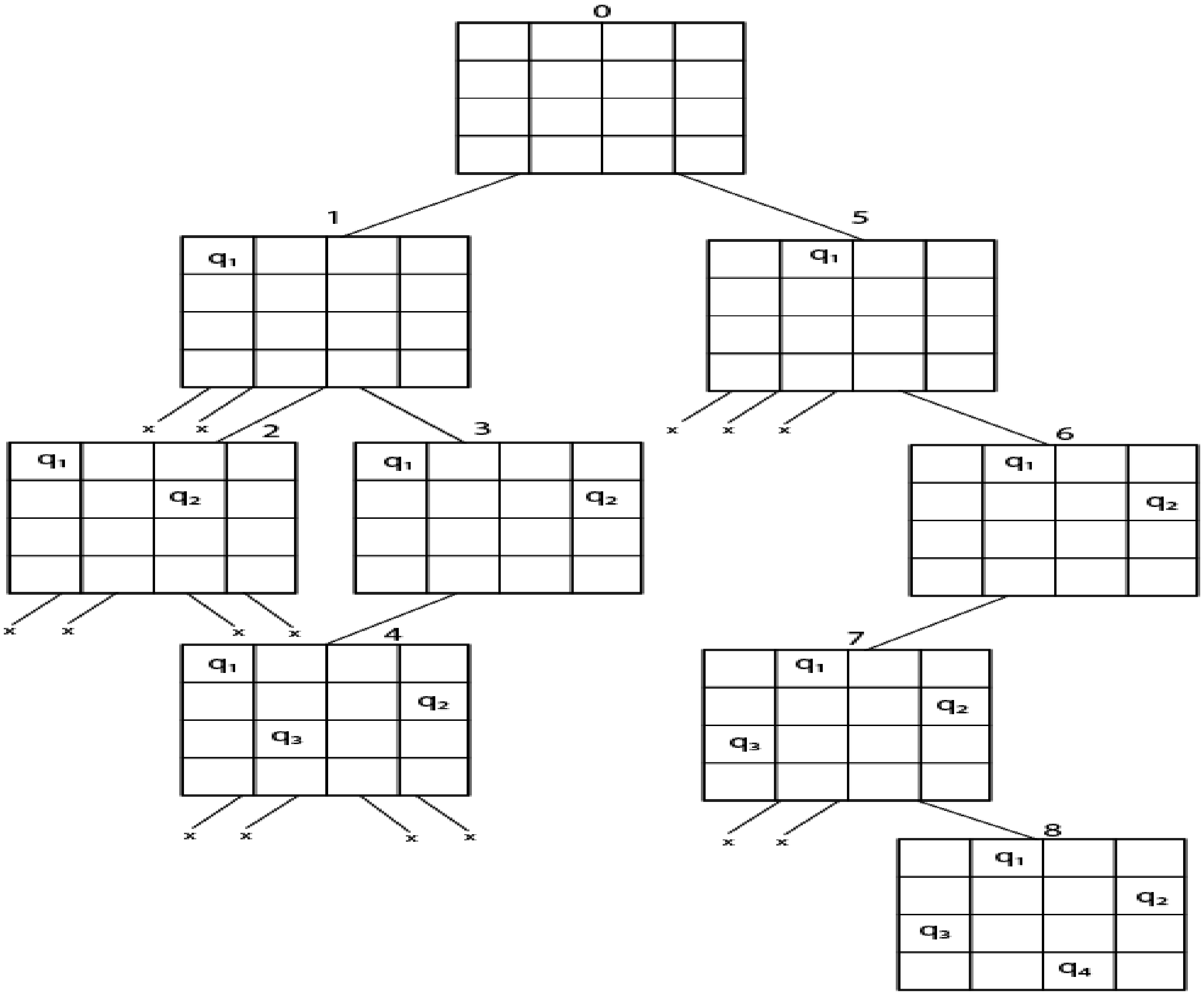
Toy Problem- 4(a)
4-queens problem

One possible solution for the 4-queens problem is (2,4,1,3) i.e ,

	1	2	3	4
1		q_1		
2				q_2
3	q_3			
4			q_4	

Toy Problem- 4(a)
4-queens problem

The implicit tree for 4 - queen problem for a solution (2, 4, 1, 3) is as follows:



Toy Problem- 4(a)
4-queens problem

Another solution for 4 - queens problems is (3, 1, 4, 2) i.e

	1	2	3	4
1			q_1	
2	q_2			
3				q_3
4		q_4		

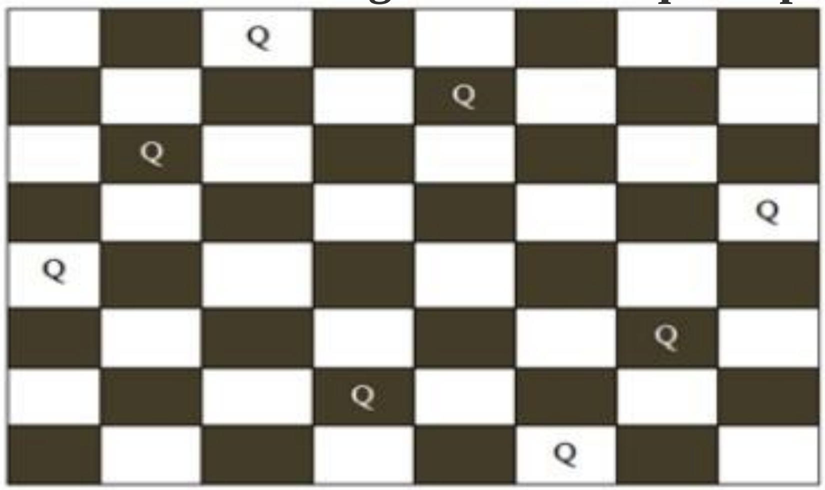
Toy Problem- 4(b)
8-queens problem

"We have 8 queens and an 8x8 Chess board having alternate black and white squares. The queens are placed on the chessboard.

Any queen can attack any other queen placed on same row, or column or diagonal. We have to find the proper placement of queens on the Chess board in such a way that no queen attacks other queen".

Toy Problem- 4(b)
8-queens problem

possible board configuration of 8 queen problem



In figure , the possible board configuration for 8-queen problem has been shown. The board has alternative black and white positions on it. The different positions on the board hold the queens. The production rule for this game is you cannot put the same queens in a same row or same column or in same diagonal. After shifting a single queen from its position on the board, the user have to shift other queens according to the production rule. Starting from the first row on the board the queen of their corresponding row and column are to be moved from their original positions to another position. Finally the player has to be ensured that no rows or columns or diagonals of on the table is same.

Toy Problem- 4(b)
8-queens problem

The first incremental formulation one might try is the following:

- **States**: Any arrangement of 0 to 8 queens on the board is a state.
- **Initial state**: No queens on the board.
- **Actions/Successor function** : Add a queen to any empty square.
- **Transition model**: Returns the board with a queen added to the specified square.
- **Goal test**: 8 queens are on the board, none attacked.
- **Path cost:** Zero (search cost only exists)

In this formulation, we have $64 \cdot 63 \cdot \cdot \cdot 57 \approx 1.8\times10^{14}$ possible sequences to investigate.

Toy Problem- 5
BLOCK WORLD

What is the Blocks World? -- The world consists of:

- A flat surface such as a tabletop
- An adequate set of identical blocks which are identified by letters.
- The blocks can be stacked one on one to form towers of apparently unlimited height.
- The stacking is achieved using a robot arm which has fundamental operations and states which can be assessed using logic and combined using logical operations.
- The robot can hold one block at a time and only one block can be moved at a time.

Toy Problem- 5

Blocks world

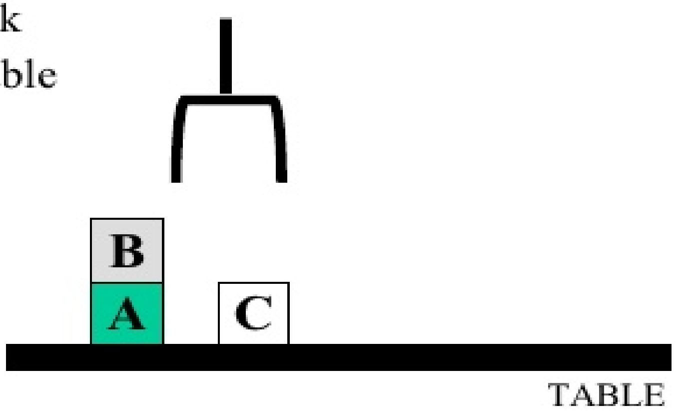
The **blocks world** is a micro-world that consists of a table, a set of blocks and a robot hand.

Some domain constraints:

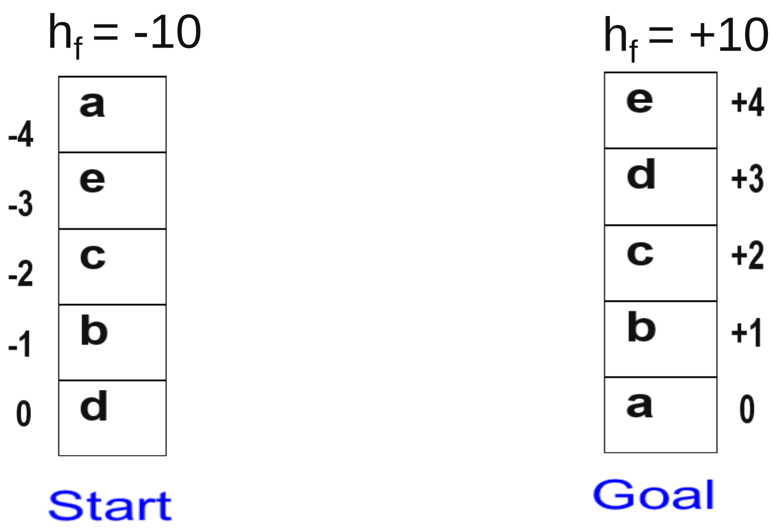
- Only one block can be on another block
- Any number of blocks can be on the table
- The hand can only hold one block

Typical representation:

ontable(a)
ontable(c)
on(b,a)
handempty
clear(b)
clear(c)



Toy Problem- 5
Blocks World Problem – Ex .



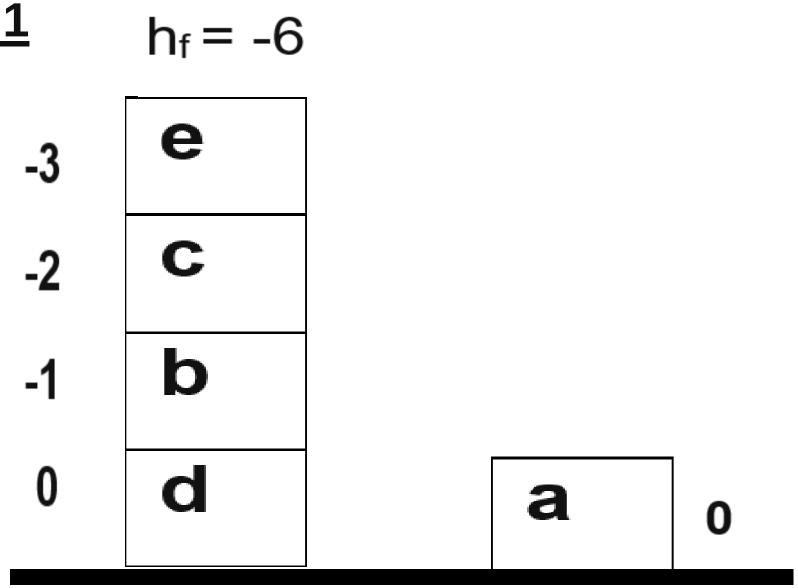
Heuristic

For each block that has the correct support structure: +1 to every block in the support structure.

For each block that has a wrong support structure: -1 to every block in the support structure.

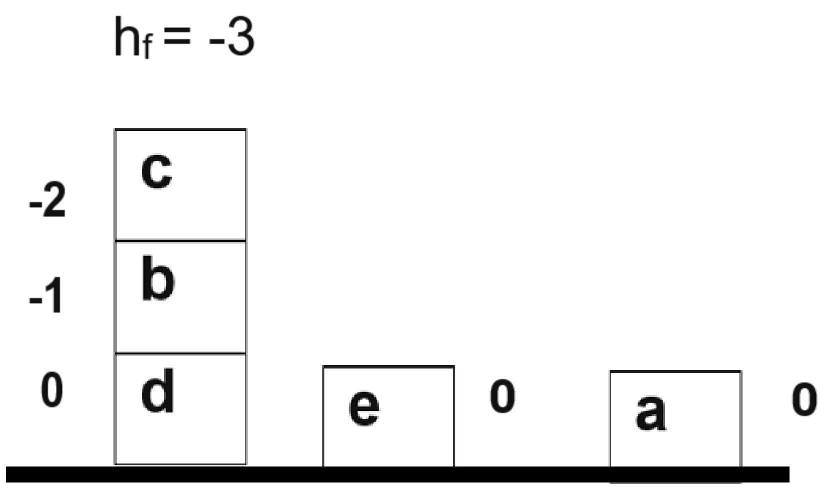
Toy Problem- 5
Blocks World Problem – Ex .

Step 1



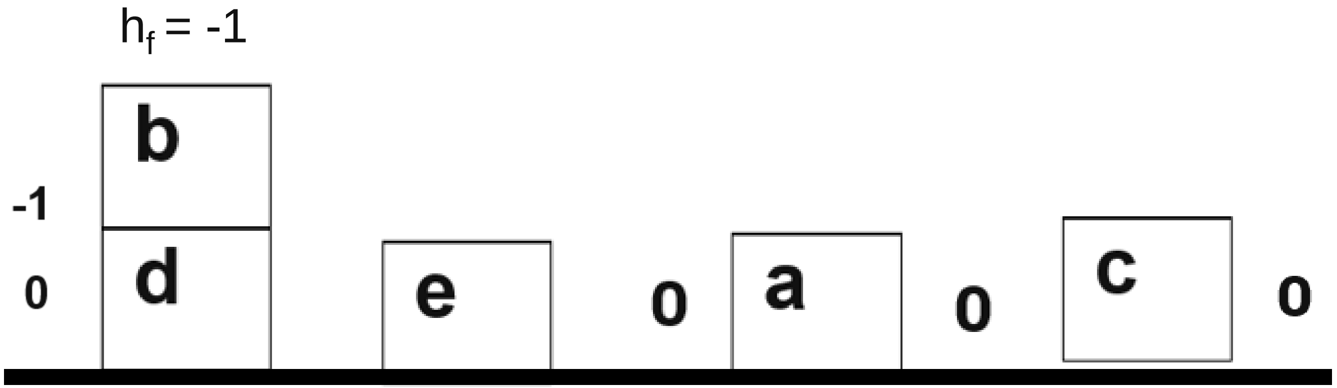
Toy Problem- 5
Blocks World Problem – Ex .

Step 2



Toy Problem- 5
Blocks World Problem – Ex .

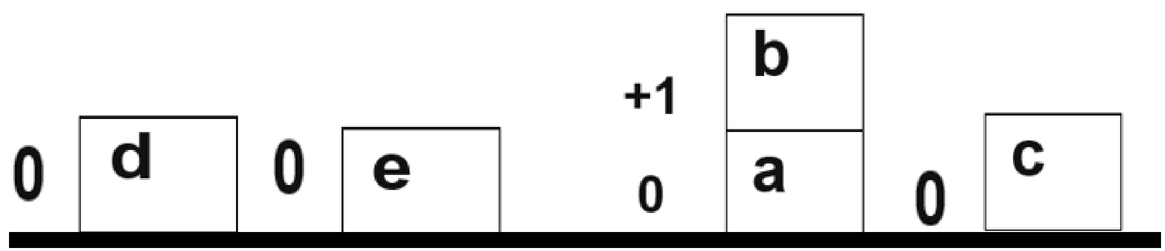
Step 3



Toy Problem- 5
Blocks World Problem – Ex .

Step 4

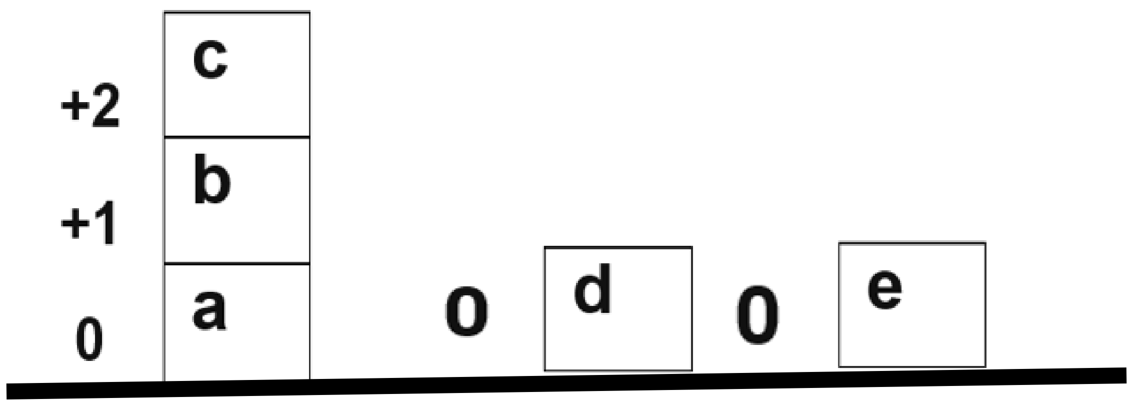
$h_f = +1$



Toy Problem- 5
Blocks World Problem – Ex .

Step 5

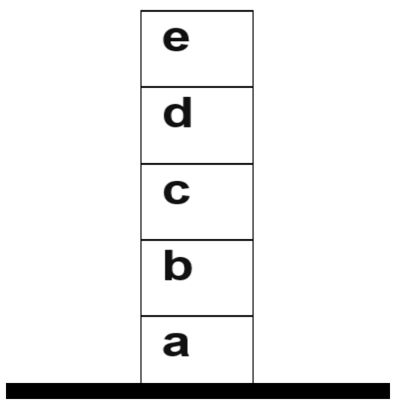
$h_f = +3$



Toy Problem- 5
Blocks World Problem – Ex .

Step 6

$h_f = +10$



Global solution for block world

Toy Problem- 5

BLOCK WORLD - STRIPS

(STanford Research Institute Problem Solver)

- STRIPS - an action-centric representation ,for each action , specifies the effect of an action.

A **STRIPS planning problem** specifies:

1) an initial state *S*
2) a goal *G*
3) a set of STRIPS actions

Toy Problem- 5

BLOCK WORLD - STRIPS

(STanford Research Institute Problem Solver)

- STRIPS - an action-centric representation ,for each action , specifies the effect of an action.

The STRIPS representation for an action consists of three lists,

- Pre_Cond list contains predicates which have to be true before operation.
- ADD list contains those predicates which will be true after operation
- DELETE list contain those predicates which are no longer true after operation

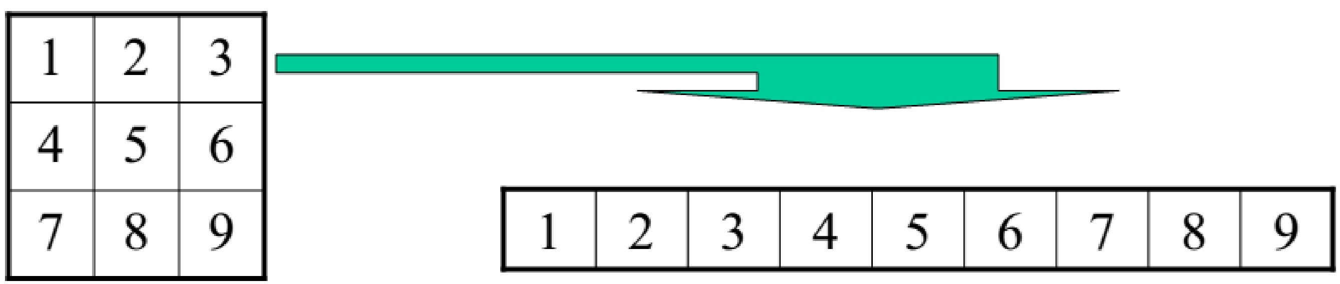
Toy Problem- 6
Tic Tac Toe

The game **Tic Tac Toe** is also known as **Noughts** and **Crosses** or **X**s and **O**s ,the player needs to take turns marking the spaces in a 3x3 grid with their own marks,
if 3 consecutive marks (**Horizontal**, **Vertical**,**Diagonal**) are formed then the player who owns these moves get won.

Assume ,

Player 1 - X
Player 2 - O

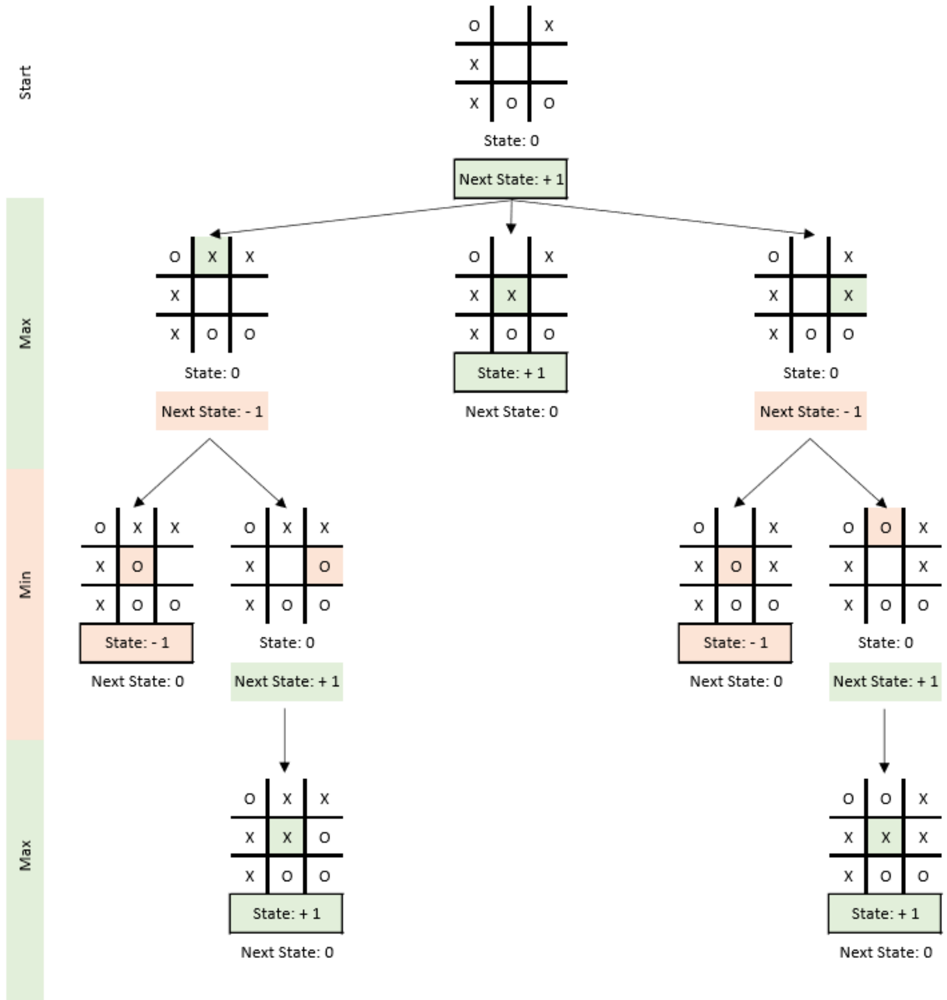
So,a player who gets 3 consecutive marks first,they will win the game .



2-D game-board

1-D Vector

Toy Problem- 6
Tic Tac Toe



Toy Problem- 7
Missionaries and Cannibals

Let Missionary is denoted by 'M' and Cannibal, by 'C'.
These rules are described below:

Rule 1 :	(0, M)	:	One missionary sailing the boat from bank-1 to bank-2
Rule 2 :	(M, 0)	:	One missionary sailing the boat from bank-2 to bank-1
Rule 3 :	(M, M)	:	Two missionaries sailing the boat from bank-1 to bank-2
Rule 4 :	(M, M)	:	Two missionaries sailing the boat from bank-2 to bank-1
Rule 5 :	(M, C)	:	One missionary and one Cannibal sailing the boat from bank-1 to bank-2
Rule 6 :	(C, M)	:	One missionary and one Cannibal sailing the boat from bank-2 to bank-1
Rule 7 :	(C, C)	:	Two Cannibals sailing the boat from bank-1 to bank-2
Rule 8 :	(C, C)	:	Two Cannibals sailing the boat from bank-2 to bank-1
Rule 9 :	(0, C)	:	One Cannibal sailing the boat from bank-1 to bank-2
Rule 10 :	(C, 0)	:	One Cannibal sailing the boat from bank-2 to bank-1

All or some of these production rules will have to be used in a particular sequence to find the solution of the problem.

Toy Problem- 7
Missionaries and Cannibals

Rules applied and their sequence in Missionaries and Cannibals problem

After application of rule	persons in the river bank-1	persons in the river bank-2	boat position
Start state	M, M, M, C, C, C	0	bank-1
5	M, M, C, C	M, C	bank-2
2	M, M, C, C, M	C	bank-1
7	M, M, M	C, C, C	bank-2
10	M, M, M, C	C, C	bank-1
3	M, C	C, C, M, M	bank-2
6	M, C, C, M	C, M	bank-1
3	C, C	C, M, M, M	bank-2
10	C, C, C	M, M, M	bank-1
7	C	M, M, M, C, C	bank-2
10	C, C	M, M, M, C	bank-1
7	0	M, M, M, C, C, C	bank-2

Toy Problem- 7

Formalization of the M&C Problem

State space: triple (x,y,z) with $0 \le x,y,z \le 3$, where x,y, and z represent the number of missionaries, cannibals and boats currently on the original bank.

Initial State: (3,3,1)

Successor function: From each state, either bring one missionary, one cannibal, two missionaries, two cannibals, or one of each type to the other bank.

Note: Not all states are attainable (e.g., (0,0,1)), and some are illegal.

Goal State: (0,0,0) Path Costs: 1 unit per crossing

Toy Problem- 8
Travelling Salesman Problem(Path Finding Problems)

"Given 'n' cities connected by roads, and distances between each pair of cities. A sales person is required to travel each of the cities exactly once. We are required to find the route of salesperson so that by covering minimum distance, he can travel all the cities and come back to the city from where the journey was started".

Diagrammatically, it is shown below

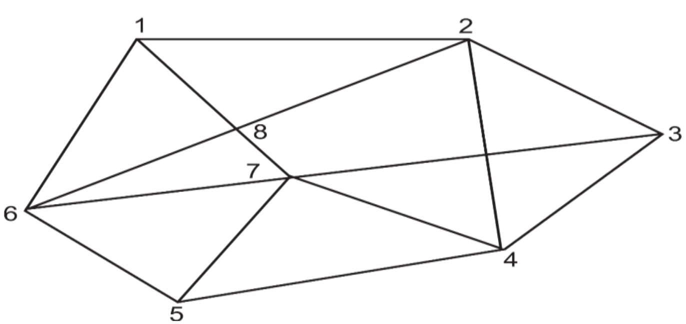


Fig : Cities and paths connecting these

Toy Problem- 8
Travelling Salesman Problem(Path Finding Problems)

The basic travelling salesperson problem comprises of computing the shortest route through a given set of cities.

Following Table shows number of cities and the possible routes mentioned against them.

Number of cities	Possible Routes
1	1
2	1 -2-1
3	1 -2 -3 1
	1 -3 -2 1
4	1- 2- 3- 4-1
	1- 2- 4- 3- 1
	1- 3- 2- 4- 1
	1- 3- 4- 2- 1
	1- 4- 2- 3-1
	1- 4- 3- 2- 1