State Space

Slider Block 3 x 3

State

Let The 3 x 3 Blocks are array (SB[3][3]) And Let 0 is Space block.

If You want to swap Space Tile and Another block just change their value each other , such as

SB[3][3] = {{5,2,7},{8,4,0},{1,3,6}} You want swap 0 and 7 then

SB[3][3] = {{5,2,0},{8,4,7},{1,3,6}}

Initial State

5	2	7
8	4	
1	3	6

*Following Input

SB[3][3] = {{5,2,7},{8,4,0},{1,3,6}}

Goal State

1	2	3
4	5	6
7	8	

SB[3][3] = {{1,2,3},{4,5,6},{7,8,0}}

Rules And Operation

1. You can move 1 Space block at one time only.

2. There are 3 Cases of movement include (Let U is Up , L is Left , R is Right and D is Down)

2.1 Space block At Corner

You can only move 2 way at one time that contain X-axis movement or Y-axis movement only.

2.2 Space block At Edge

You can only move 3 way at one time that contain 2 Y-axis and 1 X-axis movement or 2 X-axis and 1 Y-axis movement.

2.3 Space block At Middle

You can move 4 way of L R U D.

3. No swap block across another block.

Water Jug

State

Let array water[2] or water(X , Y) are operator of water jug.

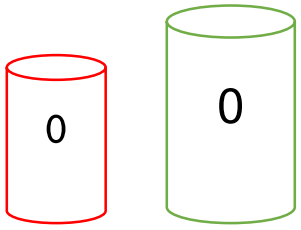
x is Volume Of Water in First Container..

y is Volume Of Water in Second Container.

Let D is Wanted Volume.

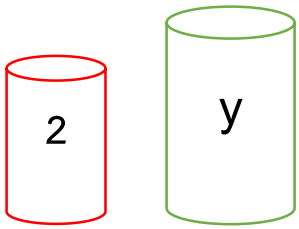
Let X is Capacity of First Container and Y is Capacity of Second Container.

Initial State



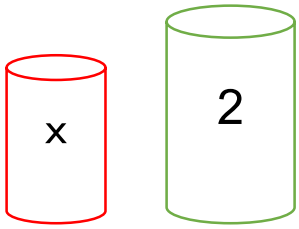
water(0 , 0)

Goal State



water(2 , y) or water(x , 2)

OR



Rules And Operation

1. Let First Container or X is only jug that can pour water to another jug.
2. If First Container or X is empty (water(0 , Y)) , then fill X (water(X , Y)).
3. If Second Container or Y is full (water(X , Y)) , then empty Y (water(X , 0)).
4. Pour between X and Y
    - 4.1 If Fill Y with X , then X is empty (water(0 , y + x)).
    - 4.2 If Fill Y with X , then X isn't empty and Y isn't full (water(X – x , y + x)).
    - 4.3 If Fill Y with X , then Y is full (water(0 , Y)).
    - 4.4 If Fill Y with X , then X isn't empty and Y is full (water(X – x , Y)).
5. You can't pour a little bit of water from any jar.

Cannibals And Missionaries

State

Let CanLeft is Amount of Cannibals at Left Side.

Let MisLeft is Amount of Missionaries at Left Side.

Let Pos is Boat position.

Let Right is Boat at Right Side.

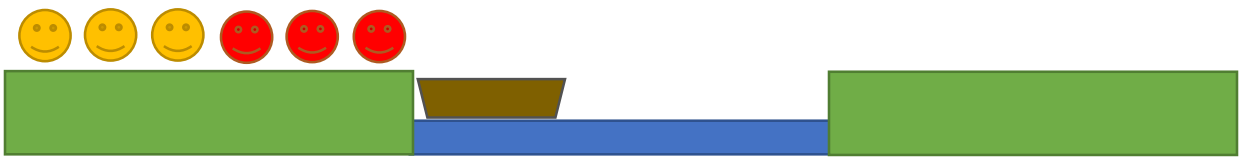
Let Left is Boat at Left Side.

Let CanRight is Amount of Cannibals at Right Side.

Let MisRight is Amount of Missionaries at Right Side.

And operation is (CanLeft , MisLeft , Pos , CanRight , MisRight)

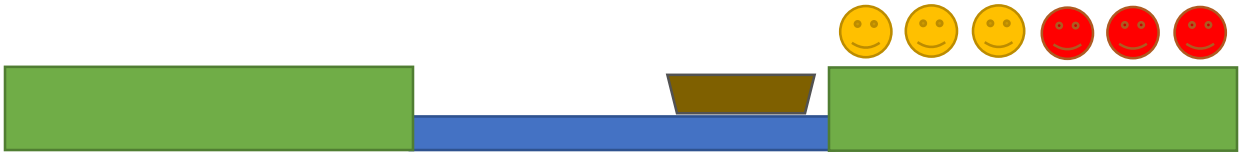
Initial State



Let There are 3 Cannibal , 3 Missionaries and 1 Boat On Left Side

(3 , 3 , Left , 0 , 0)

Goal State



Let There are 3 Cannibal , 3 Missionaries and 1 Boat On Right Side

(0 , 0 , Right , 3 , 3)

Rules And Operation

1. The Boat can carry 1 or 2 people at one time.
2. Don't let amount of cannibals more than amount of missionaries on one side.
3. If Boat is Right , then Boat can go to Left
4. If Boat is Left , then Boat can go to right
5. If Cannibal gets move , then CanLeft or CanRight decreases.
6. If Missionary gets move , then MisLeft or MisRight decreases.
7. If Cannibal is on the other land , then CanLeft OR CanRight increases.
7. If Missionary is on the other land , then MisLeft OR MisRight increases.

Knapsack

State

Let There are N Things that contain with Value (V) , Weight (W) and bitwise operation (0 or 1) by 0 is not picked and 1 is picked : N(V , W , B).

Let Knapsack Capacity is K that contain with Current Value (CV) and Current Weight (CW)

MV is Maximum Value

MW is Maximum Weight

Initial State

1(50 , 10 , 0)

2(75 , 20 , 0)

3(25 , 30 , 0)

K(0 , 0)

There are 3 Thing : 1(50 , 10 , 0) , 2(75 , 20 , 0) and 3(25 , 30 , 0)

And Knapsack Capacity is K(0 , 0)

Goal State

Knapsack Capacity is K(MV , CW ; CW <= MW)

Rules And Operation

1. If You want to pick N thing , Then N(V , W , 1) and CV+=V , CW+=W
2. If You don't want to pick N thing , Then N(V , W , 0) and CV-=V , CW-=W
3. You can't have current weight (CW) more than maximum weight of knapsack.
4. You can't pick nothing.