

## 1. Solution Approach:

I have Followed the Warnsdorff's Algorithm approach to solve the problem.

step 1: Build the 10\*10 Board with unvisited(-1)

step 2: prepare all possible patterns

row\_pattern = [3, 2, 0, -2, -3, -2, 0, 2]

col\_pattern = [0, -2, -3, -2, 0, 2, 3, 2]

step 3: Pawn can start from any valid position on the board.

step 4: By using the above pattern. get the unvisited(-1) tile with minimal degree from the current position

step 5: Once you find the next position, update the position on board as visited.

Step 6: Keep continue step 4 and 5 until all the tiles got visited

note: Board is in one dimensional array, Index calculation will be

board[col \* BOARD\_SIZE + row]

## 2. App setup and run the App:

2.1. Required ruby 2.5.1

2.2. unzip the app folder and go to the app path.

2.3. Run bundle install

**\$ bundle install**

2.4. To run the application with the command below. pass the row and col values of the initial position.

if no arguments passed by default 0,0 will be considered.

**\$ ruby lib/visit.rb 3 2**

## 3. Running Unit test cases:

**\$ rspec spec/pawn\_tour\_spec.rb**

## 4. Sample output:

**\$ ruby lib/visit.rb 3 2**

33	72	10	34	69	11	41	50	12	42
2	87	31	99	88	81	46	95	82	45
9	35	68	73	40	49	74	43	48	51
32	71	1	84	70	100	83	59	13	94
3	86	30	98	89	80	47	96	79	44
8	36	67	76	39	60	75	93	23	52
27	56	6	85	57	97	90	58	14	91
4	65	29	17	66	77	22	61	78	21
7	37	26	55	38	25	54	92	24	53
28	18	5	64	19	16	63	20	15	62

Find the attachment of the sample output in the app folder.