**Problem**

* **nodeExpanded**: total node expanded
* **startState** (*list*)
* **nonePosition**: position of “0” in startState
* **dicGoal** (*dict*): a dictionary contain position with its number

Funtion

* **findNonePosition**(*matrix*): find “0” position in matrix

+ Return a tupe (x, y)

* **setStartState**(*matrix*): set startState equal to matrix and set nonePosition through findNonePosition function
* **setGoalState**(*matrix*): set goalState equal to matrix and create dicGoal
* **getStartState**(): return a tupe (startState, nonePosition)
* **getGoalState**(): return a tupe (goalState, dicGoal)
* **isGoalState**(*state*):

+ state = (matrix, nonePosition)

+ compare matrix with matrix of goalState

* **isValidPosition**(*x, y*): if (x, y) out of matrix board return False, else return True
* **getSuccessor**(*state*): return a list contain successors of current state

+ each state have maximum 4 successor with 4 direction “North, East, South, West”

+ once getSuccessors was called, “nodeExpanded” will increase by 1

**PriorityQueu**:

**Direction**:

**Solve\_Problem**

Function

* **aStarSearch**(*problem*):
  + Run astar search in problem(class problem above) to solve it.
  + Return: a tupe (state, moveDirection, cost, nonePositionMove)

+ state = (matrix, nonePosition)

+ moveDirection: list contain direction move from start state to goal. Ex: [‘North’, ‘West’]

+ cost: total step to reach the goal

+ nonePositionMove: list contain the changing of “0” position from start state to goal

* **getAnswer**(*problem*):
  + Run aStartSearch in and return 2 value: nonePositionMove, nodeExpanded
* **printState**(*state*): print state in console
  + state = matrix
* **printGoalState**(*problem*): print goal state from problem(class problem above) in console
* **printStartState**(*problem*):
* **printAnswer**(*problem*):
  + Print answer after run aStartSearch
* **heuristicFunction**(*state, problem*):
  + Estimate the heuristic value “h(x)” from state to goal
* **doAction**(*problem*):
  + solve the problem and print state step by step