```
Total time: 9.827s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/breathFirstSearch.py
Function: expand at line 24

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    24                                                  @cpu
    25                                                  def expand(board):
    26    201468      148.0ms            .      1.5%        for i in range(len(board.data)):                               # to find the loca
    27    503660      185.8ms            .      1.9%            for j in range(len(board.data[i])):
    28    402926      169.9ms            .      1.7%                if board.data[i][j] == '*':
    29     50367       21.7ms            .      0.2%                    location = [i,j];
    30     50367       13.4ms            .      0.1%                    break
    31
    32     50367       13.7ms            .      0.1%        actions = []
    33    186814        1.08s            .     11.0%        for move in possible_actions(constants.board, location):       # to find all poss
    34    136447        8.18s        0.1ms     83.2%            actions.append([result(location, move, board.data) , move]) # prepare all poss
    35
    36     50367       14.7ms            .      0.2%        return actions                                                 # After expanding

Total time: 0.483s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/breathFirstSearch.py
Function: possible_actions at line 38

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    38                                                  @cpu
    39                                                  def possible_actions(board, location):                         # to find all poss
    40     50367       31.0ms            .      6.4%        actions = ["RIGHT","LEFT","UP","DOWN"]
    41     50367       16.2ms            .      3.3%        actionstopeform = []
    42
    43    251835       84.4ms            .     17.5%        for x in actions:
    44                                                              # for moving right
    45    201468       65.0ms            .     13.4%            if x == "RIGHT":
    46     50367       26.5ms            .      5.5%                if location[1]+1 < len(board):
    47     34145       24.5ms            .      5.1%                    actionstopeform.append([x,location[0],location[1]+1])
    48                                                              # for moving left
    49    151101       53.0ms            .     11.0%            elif x == "LEFT":
    50     50367       21.9ms            .      4.5%                if location[1]-1 >= 0:
    51     34135       21.4ms            .      4.4%                    actionstopeform.append([x,location[0],location[1]-1])
    52                                                              # for moving up
    53    100734       30.8ms            .      6.4%            elif x == "UP":
    54     50367       20.1ms            .      4.2%                if location[0]-1 >= 0:
    55     34084       18.0ms            .      3.7%                    actionstopeform.append([x,location[0]-1,location[1]])
    56                                                              # for moving down
    57     50367       15.2ms            .      3.2%            elif x == "DOWN":
    58     50367       22.7ms            .      4.7%                if location[0]+1 < len(board):
    59     34083       17.9ms            .      3.7%                    actionstopeform.append([x,location[0]+1,location[1]])
    60
    61     50367       14.5ms            .      3.0%        return actionstopeform

Total time: 7.716s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/breathFirstSearch.py
Function: result at line 63

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    63                                                  @cpu
    64                                                  def result(location,action,board):
    65    136447        6.42s            .     83.1%        newBoard = copy.deepcopy(board)                           # copy of a board so t
    66    136447      431.1ms            .      5.6%        temp = copy.deepcopy(newBoard[action[1]][action[2]])
    67    136447      423.7ms            .      5.5%        newBoard[action[1]][action[2]] = copy.deepcopy('*')
    68    136447      409.8ms            .      5.3%        newBoard[location[0]][location[1]]  = copy.deepcopy(temp)
    69    136447       36.3ms            .      0.5%        return newBoard                                           # return new board aft

Total time: 254.715s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/breathFirstSearch.py
Function: bfs at line 71

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    71                                                  @memory_profiler.profile
    72                                                  @cpu
    73                                                  def bfs(board):                                           # function BREADTH-FIR
    74         1            .            .         .        frontier = queue.Queue()
    75         1            .            .         .        node = Node(data = board)                             # node ← NODE(problem.
    76         1            .            .         .        frontier.put(node)                                    # frontier ← a FIFO qu
    77                                                      # maxQueueSize = 1                                    # only for debug
    78         1            .            .         .        if constants.goalBoard == node.data:                  # if problem.IS-GOAL(n
    79                                                          return node                                       # then return node
    80         1            .            .         .        reached = []
    81         1            .            .         .        reached.append(board)                                 # reached ← {problem.I
    82     50367      290.2ms            .      0.1%        while not frontier.empty():                           # while not IS-EMPTY(f
    83     50367      518.4ms            .      0.2%            val = frontier.get()                              # node ← POP(frontier
    84    186812       10.67s        0.1ms      4.2%            for child in expand(val):                         # for each child in EX
    85    136446      651.6ms            .      0.3%                s =  Node(data=child[0], depth = val.depth + 1, move= child[1] , prev=val)
    86    136446      118.6ms            .         .                if goalBoard == s.data:                       # if problem.IS-GOAL(s
    87                                                                  #print('Max queue size:', maxQueueSize)   # only for debug
    88         1            .            .         .                    return s                                  # then return child
    89    136445      240.97s        1.8ms     94.6%                if s.data not in reached:                     # if s is not in reach
    90     68948      211.5ms            .      0.1%                    reached.append(s.data)                    # add s to reached
    91     68948        1.28s            .      0.5%                    frontier.put(s)                           # add child to frontie
    92                                                                  # maxQueueSize+=1                         # only for debug
    93                                                          #print('Max queue size:', maxQueueSize)            # only for debug
    94                                                        return failure                                                # return failure


Total time: 0.001s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/breathFirstSearch.py
Function: printStatistics at line 96

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    96                                                  @cpu
    97                                                  def printStatistics(solution):
    98         1            .            .      0.3%        pathCost = 0
    99         1            .            .      0.1%        stateSequence = []
   100         1            .            .      0.1%        actionSequence = []
   101
   102        22            .            .      2.3%        while solution.prev != None:
   103        21            .            .      1.4%            stateSequence.insert(0, solution.data)
   104        21            .            .      1.1%            actionSequence.insert(0, solution.move)
   105        21            .            .      1.1%            solution = solution.prev
   106        21            .            .      1.4%            pathCost += 1
   107
   108         1            .            .      5.1%        print('Action sequence:')
   109         1        0.1ms        0.1ms     13.9%        print(*actionSequence, sep='\n')
   110
   111         1            .            .      1.4%        print('\nState sequence:')
   112         1        0.5ms        0.5ms     69.3%        print(*stateSequence, sep='\n')
   113
   114         1            .            .      2.6%        print('\nPath cost:', pathCost)
```