```
Total time: 0.007s
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       208       0.1ms            .      0.8%        for i in range(len(board.data)):                                # to find the loca
    27       520       0.1ms            .      2.0%            for j in range(len(board.data[i])):
    28       416       0.1ms            .      1.6%                if board.data[i][j] == '*':
    29        52           .            .      0.2%                    location = [i,j];
    30        52           .            .      0.1%                    break
    31
    32        52           .            .      0.2%        actions = []
    33       201       0.8ms            .     10.7%        for move in possible_actions(constants.board, location):        # to find all poss
    34       149       5.9ms            .     84.2%            actions.append([result(location, move, board.data) , move]) # prepare all poss
    35
    36        52           .            .      0.2%        return actions                                                  # After expanding


Total time: 0.000s
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        52           .            .      4.4%        actions = ["RIGHT","LEFT","UP","DOWN"]
    41        52           .            .      3.1%        actionstopeform = []
    42
    43       260       0.1ms            .     16.0%        for x in actions:
    44                                                       # for moving right
    45       208           .            .     13.8%            if x == "RIGHT":
    46        52           .            .      4.4%                if location[1]+1 < len(board):
    47        35           .            .      3.1%                    actionstopeform.append([x,location[0],location[1]+1])
    48                                                       # for moving left
    49       156           .            .      9.4%            elif x == "LEFT":
    50        52           .            .      4.7%                if location[1]-1 >= 0:
    51        35           .            .      3.8%                    actionstopeform.append([x,location[0],location[1]-1])
    52                                                       # for moving up
    53       104           .            .      8.2%            elif x == "UP":
    54        52           .            .      4.1%                if location[0]-1 >= 0:
    55        40           .            .      5.0%                    actionstopeform.append([x,location[0]-1,location[1]])
    56                                                       # for moving down
    57        52           .            .      4.4%            elif x == "DOWN":
    58        52           .            .      6.3%                if location[0]+1 < len(board):
    59        39           .            .      5.3%                    actionstopeform.append([x,location[0]+1,location[1]])
    60
    61        52           .            .      4.1%        return actionstopeform


Total time: 0.006s
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       149       4.6ms            .     81.0%        newBoard = copy.deepcopy(board)                                 # copy of a board so t
    66       149       0.3ms            .      6.1%        temp = copy.deepcopy(newBoard[action[1]][action[2]])
    67       149       0.4ms            .      6.4%        newBoard[action[1]][action[2]] = copy.deepcopy('*')
    68       149       0.3ms            .      5.9%        newBoard[location[0]][location[1]]  = copy.deepcopy(temp)
    69       149           .            .      0.6%        return newBoard                                                 # return new board aft


Total time: 0.009s
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-FIF
    74         1           .            .      0.2%        frontier = queue.Queue()
    75         1           .            .        .         node = Node(data = board)                                       # node ← NODE(problem.
    76         1           .            .      0.1%        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        52       0.1ms            .      0.6%        while not frontier.empty():                                     # while not IS-EMPTY(f
    83        52       0.2ms            .      2.3%            val = frontier.get()                                        # node ← POP(frontier
    84       199       7.5ms            .     87.3%            for child in expand(val):                                   # for each child in EX
    85       148       0.2ms            .      2.0%                s =  Node(data=child[0], depth = val.depth + 1, move= child[1] , prev=val)
    86       148       0.1ms            .      0.6%                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       147       0.2ms            .      2.7%                if s.data not in reached:                               # if s is not in reach
    90        96           .            .      0.5%                    reached.append(s.data)                              # add s to reached
    91        96       0.3ms            .      3.6%                    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: 0.000s
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            .            .      2.2%        pathCost = 0
    99         1            .            .         .        stateSequence = []
   100         1            .            .         .        actionSequence = []
   101
   102         7            .            .      4.4%        while solution.prev != None:
   103         6            .            .      4.4%            stateSequence.insert(0, solution.data)
   104         6            .            .      6.7%            actionSequence.insert(0, solution.move)
   105         6            .            .         .            solution = solution.prev
   106         6            .            .      4.4%            pathCost += 1
   107
   108         1            .            .      8.9%        print('Action sequence:')
   109         1            .            .     28.9%        print(*actionSequence, sep='\n')
   110
   111         1            .            .      4.4%        print('\nState sequence:')
   112         1            .            .     28.9%        print(*stateSequence, sep='\n')
   113
   114         1            .            .      6.7%        print('\nPath cost:', pathCost)
```