```
Total time: 14.403s
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    335432      221.4ms            .      1.5%        for i in range(len(board.data)):                                  # to find the loca
    27    838579      286.9ms            .      2.0%            for j in range(len(board.data[i])):
    28    670863      244.1ms            .      1.7%                if board.data[i][j] == '*':
    29     83858       33.8ms            .      0.2%                    location = [i,j];
    30     83858       22.2ms            .      0.2%                    break
    31
    32     83858       21.3ms            .      0.1%        actions = []
    33    308279        1.61s            .     11.2%        for move in possible_actions(constants.board, location):         # to find all poss
    34    224421       11.94s        0.1ms     82.9%            actions.append([result(location, move, board.data) , move]) # prepare all poss
    35
    36     83858       22.4ms            .      0.2%        return actions                                                    # After expanding


Total time: 0.743s
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     83858       46.6ms            .      6.3%        actions = ["RIGHT","LEFT","UP","DOWN"]
    41     83858       25.6ms            .      3.4%        actionstopeform = []
    42
    43    419290      129.9ms            .     17.5%        for x in actions:
    44                                                           # for moving right
    45    335432       98.6ms            .     13.3%            if x == "RIGHT":
    46     83858       40.1ms            .      5.4%                if location[1]+1 < len(board):
    47     56129       37.4ms            .      5.0%                    actionstopeform.append([x,location[0],location[1]+1])
    48                                                           # for moving left
    49    251574       82.6ms            .     11.1%            elif x == "LEFT":
    50     83858       34.1ms            .      4.6%                if location[1]-1 >= 0:
    51     56128       32.3ms            .      4.3%                    actionstopeform.append([x,location[0],location[1]-1])
    52                                                           # for moving up
    53    167716       49.2ms            .      6.6%            elif x == "UP":
    54     83858       31.3ms            .      4.2%                if location[0]-1 >= 0:
    55     56070       27.0ms            .      3.6%                    actionstopeform.append([x,location[0]-1,location[1]])
    56                                                           # for moving down
    57     83858       24.4ms            .      3.3%            elif x == "DOWN":
    58     83858       34.0ms            .      4.6%                if location[0]+1 < len(board):
    59     56094       27.5ms            .      3.7%                    actionstopeform.append([x,location[0]+1,location[1]])
    60
    61     83858       22.6ms            .      3.0%        return actionstopeform


Total time: 11.276s
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    224421        9.35s            .     82.9%        newBoard = copy.deepcopy(board)                                 # copy of a board so t
    66    224421      646.8ms            .      5.7%        temp = copy.deepcopy(newBoard[action[1]][action[2]])
    67    224421      620.4ms            .      5.5%        newBoard[action[1]][action[2]] = copy.deepcopy('*')
    68    224421      602.7ms            .      5.3%        newBoard[location[0]][location[1]]  = copy.deepcopy(temp)
    69    224421       57.3ms            .      0.5%        return newBoard                                                 # return new board aft


Total time: 533.657s
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            .            .         .        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     83858      414.3ms            .      0.1%        while not frontier.empty():                                     # while not IS-EMPTY(f
    83     83858      750.6ms            .      0.1%            val = frontier.get()                                        # node ← POP(frontier
    84    308277       15.64s        0.1ms      2.9%            for child in expand(val):                                   # for each child in EX
    85    224420      862.7ms            .      0.2%                s =  Node(data=child[0], depth = val.depth + 1, move= child[1] , prev=val)
    86    224420      191.0ms            .         .            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    224419      513.97s        2.3ms     96.3%            if s.data not in reached:                                   # if s is not in reach
    90    105271      264.0ms            .         .                reached.append(s.data)                                  # add s to reached
    91    105271        1.56s            .      0.3%                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            .            .      0.5%        pathCost = 0
    99         1            .            .      0.2%        stateSequence = []
   100         1            .            .         .        actionSequence = []
   101
   102        24            .            .      1.7%        while solution.prev != None:
   103        23            .            .      3.0%            stateSequence.insert(0, solution.data)
   104        23            .            .      2.0%            actionSequence.insert(0, solution.move)
   105        23            .            .      1.7%            solution = solution.prev
   106        23            .            .      2.2%            pathCost += 1
   107
   108         1            .            .      6.9%        print('Action sequence:')
   109         1        0.1ms        0.1ms     16.8%        print(*actionSequence, sep='\n')
   110
   111         1        0.2ms        0.2ms     46.8%        print('\nState sequence:')
   112         1        0.1ms        0.1ms     13.9%        print(*stateSequence, sep='\n')
   113
   114         1            .            .      4.2%        print('\nPath cost:', pathCost)
```