```
Total time: 0.007s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: expand at line 61

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
    61                                                  @cpu
    62                                                  def expand(board):
    63       208        0.1ms            .      1.0%        for i in range(len(board.data)):
    64       520        0.1ms            .      1.8%            for j in range(len(board.data[i])):
    65       416        0.1ms            .      1.8%                if board.data[i][j] == '*':
    66        52            .            .      0.2%                    location = [i,j];
    67        52            .            .      0.1%                    break
    68
    69        52            .            .      0.1%        actions = []
    70       201        0.8ms            .     10.1%        for move in possible_actions(constants.board, location):
    71       149        6.3ms            .     84.8%            actions.append([result(location, move, board.data) , move])
    72
    73        52            .            .      0.1%        return actions


Total time: 0.000s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: possible_actions at line 75

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
    75                                                  @cpu
    76                                                  def possible_actions(board, location):
    77        52            .            .      3.4%        actions = ["RIGHT","LEFT","UP","DOWN"]
    78        52            .            .      4.5%        actionstopeform = []
    79
    80       260        0.1ms            .     14.0%        for x in actions:
    81                                                          # for moving right
    82       208        0.1ms            .     15.9%            if x == "RIGHT":
    83        52            .            .      4.5%                if location[1]+1 < len(board):
    84        35            .            .      5.3%                    actionstopeform.append([x,location[0],location[1]+1])
    85                                                          # for moving left
    86       156            .            .     11.1%            elif x == "LEFT":
    87        52            .            .      5.3%                if location[1]-1 >= 0:
    88        35            .            .      2.9%                    actionstopeform.append([x,location[0],location[1]-1])
    89                                                          # for moving up
    90       104            .            .      6.1%            elif x == "UP":
    91        52            .            .      4.8%                if location[0]-1 >= 0:
    92        40            .            .      4.0%                    actionstopeform.append([x,location[0]-1,location[1]])
    93                                                          # for moving down
    94        52            .            .      5.0%            elif x == "DOWN":
    95        52            .            .      3.7%                if location[0]+1 < len(board):
    96        39            .            .      5.6%                    actionstopeform.append([x,location[0]+1,location[1]])
    97
    98        52            .            .      4.0%        return actionstopeform


Total time: 0.006s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: result at line 100

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
   100                                                  @cpu
   101                                                  def result(location,action,board):
   102                                                      # copy of a board so that we can modify it
   103       149        4.9ms            .     81.6%        newBoard = copy.deepcopy(board)
   104       149        0.4ms            .      6.1%        temp = copy.deepcopy(newBoard[action[1]][action[2]])
   105       149        0.4ms            .      5.9%        newBoard[action[1]][action[2]] = copy.deepcopy('*')
   106       149        0.3ms            .      5.8%        newBoard[location[0]][location[1]]  = copy.deepcopy(temp)
   107                                                      # return new board after moving * - NIL to the new location
   108       149            .            .      0.5%        return newBoard


Total time: 0.009s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: bfs at line 173

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
   173                                                  @cpu
   174                                                  def bfs(board):
   175         1            .            .      0.2%        frontier = queue.Queue()
   176         1            .            .         .        node = Node(data = board)
   177         1            .            .      0.1%        frontier.put(node)
   178         1            .            .         .        maxQueueSize = 1
   179         1            .            .         .        if constants.goalBoard == node.data:
   180                                                          print('mill gya:', maxQueueSize)
   181                                                          return node
   182
   183         1            .            .         .        reached = []
   184         1            .            .         .        reached.append(board)
   185
   186        52        0.1ms            .      0.7%        while not frontier.empty():
   187        52        0.2ms            .      2.1%            val = frontier.get()
   188       199        7.9ms            .     86.9%            for child in expand(val):
   189       148        0.2ms            .      2.1%                s =  Node(data=child[0], depth = val.depth + 1, move= child[1] , prev=val)
   190
   191       148        0.1ms            .      0.6%                if goalBoard == s.data:
   192         1            .            .      0.2%                    print('Max queue size:', maxQueueSize)
   193         1            .            .         .                    return s
   194       147        0.2ms            .      2.6%                if s.data not in reached:
```

```
195        96            .            .      0.5%                    reached.append(s.data)
196        96        0.3ms            .      3.6%                    frontier.put(s)
197        96            .            .      0.4%                    maxQueueSize+=1
198
199                                                          print('Max queue size:', maxQueueSize)
200
201                                                          return failure
```