```
Total time: 0.035s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: expand at line 48

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
    48                                                  @cpu
    49                                                  def expand(board):
    50       988        0.3ms            .      0.8%        for i in range(len(board.data)):
    51      2450        0.7ms            .      2.0%            for j in range(len(board.data[i])):
    52      1956        0.6ms            .      1.7%                if board.data[i][j] == '*':
    53       247        0.1ms            .      0.2%                    location = [i,j];
    54       247            .            .      0.1%                    break
    55
    56       247        0.1ms            .      0.2%        actions = []
    57       928        3.7ms            .     10.6%        for move in possible_actions(constants.board, location):
    58       681       29.6ms            .     84.3%            actions.append([result(location, move, board.data) , move])
    59
    60       247        0.1ms            .      0.1%        return actions


Total time: 0.002s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: possible_actions at line 62

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
    62                                                  @cpu
    63                                                  def possible_actions(board, location):
    64       247        0.1ms            .      4.2%        actions = ["RIGHT","LEFT","UP","DOWN"]
    65       247        0.1ms            .      3.9%        actionstopeform = []
    66
    67      1235        0.3ms            .     19.2%        for x in actions:
    68                                                          # for moving right
    69       988        0.2ms            .     14.1%            if x == "RIGHT":
    70       247        0.1ms            .      5.6%                if location[1]+1 < len(board):
    71       181        0.1ms            .      4.1%                    actionstopeform.append([x,location[0],location[1]+1])
    72                                                          # for moving left
    73       741        0.2ms            .     10.6%            elif x == "LEFT":
    74       247        0.1ms            .      4.8%                if location[1]-1 >= 0:
    75       161        0.1ms            .      3.4%                    actionstopeform.append([x,location[0],location[1]-1])
    76                                                          # for moving up
    77       494        0.1ms            .      6.6%            elif x == "UP":
    78       247        0.1ms            .      4.6%                if location[0]-1 >= 0:
    79       172        0.1ms            .      3.9%                    actionstopeform.append([x,location[0]-1,location[1]])
    80                                                          # for moving down
    81       247        0.1ms            .      3.3%            elif x == "DOWN":
    82       247        0.1ms            .      5.0%                if location[0]+1 < len(board):
    83       167        0.1ms            .      3.6%                    actionstopeform.append([x,location[0]+1,location[1]])
    84
    85       247        0.1ms            .      3.0%        return actionstopeform


Total time: 0.028s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: result at line 87

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
    87                                                  @cpu
    88                                                  def result(location,action,board):
    89                                                      # copy of a board so that we can modify it
    90       681       22.8ms            .     81.5%        newBoard = copy.deepcopy(board)
    91       681        1.7ms            .      6.1%        temp = copy.deepcopy(newBoard[action[1]][action[2]])
    92       681        1.7ms            .      6.0%        newBoard[action[1]][action[2]] = copy.deepcopy('*')
    93       681        1.6ms            .      5.8%        newBoard[location[0]][location[1]]  = copy.deepcopy(temp)
    94                                                      # return new board after moving * - NIL to the new location
    95       681        0.2ms            .      0.6%        return newBoard


Total time: 0.004s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: misplaced at line 170

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
   170                                                  @cpu
   171                                                  def misplaced(puzzle):
   172       413        0.1ms            .      2.3%        num_misplaced = 0
   173      1652        0.5ms            .     11.6%        for i in range(len(puzzle.data)):
   174      4956        1.3ms            .     31.8%            for j in range(len(puzzle.data)):
   175      3717        1.6ms            .     39.4%                if puzzle.data[i][j] != constants.goalBoard[i][j] and puzzle.data[i][j] !=
   176      2134        0.5ms            .     12.7%                    num_misplaced += 1
   177       413        0.1ms            .      2.2%        return num_misplaced


Total time: 0.051s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: a_star at line 235

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
   235                                                  @cpu
   236                                                  def a_star(initialProblem, f):
   237         1            .            .         .        initialNode = Node(data = initialProblem)   # node←NODE(STATE=problem.INITIAL)
   238         1            .            .         .        frontier = PriorityQueue()
   239         1            .            .         .        frontier.append((f(initialNode), initialNode))         # frontier←a priority queue
   240
```

```
   241         1            .            .         .        reached = {str(initialProblem): initialNode}           # reached←a lookup table, w
   242
   243       248        0.2ms            .      0.3%        while not frontier.empty():                            # while not IS-EMPTY(fronti
   244       248        0.2ms            .      0.3%            node = frontier.get()                              # node←POP(frontier)
   245
   246       248        0.1ms            .      0.3%            if constants.goalBoard == node[1].data:            # if problem.IS-GOAL(node.S
   247         1            .            .         .                print('Max queue size:', frontier.getSize())
   248         1            .            .         .                return node[1]                                 # then return node
   249
   250       928       37.3ms            .     72.5%            for child in expand(node[1]):                # for each child in EXPAND(problem
   251                                                               # s←child.STATE
   252       681        0.9ms            .      1.8%               s =  Node( data = child[0], depth = node[1].depth + 1, move = child[1], pr
   253
   254                                                               # if s is not in reached or child.PATH-COST < reached[s].PATH-COST then
   255       681        1.4ms            .      2.7%               if str(s.data) not in reached or s.depth < reached[str(s.data)].depth:
   256       412        0.6ms            .      1.1%                   reached[str(s.data)] = s                    # reached[s]←child
   257       412       10.7ms            .     20.9%                   frontier.append((f(s) ,s))                  # add child to frontier
   258
   259                                                           return constants.failure                          # return failure


Total time: 0.000s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py
Function: printStatistics at line 261

Line #      Hits         Time      Per Hit    % Time    Line Contents
==============================================================
   261                                                  @cpu
   262                                                  def printStatistics(solution):
   263         1            .            .      0.5%        pathCost = 0
   264         1            .            .      0.5%        stateSequence = []
   265         1            .            .         .        actionSequence = []
   266
   267        34            .            .      5.9%        while solution.prev != None:
   268        33            .            .      6.5%            stateSequence.insert(0, solution.data)
   269        33            .            .      4.8%            actionSequence.insert(0, solution.move)
   270        33            .            .      6.5%            solution = solution.prev
   271        33            .            .      3.2%            pathCost += 1
   272
   273         1            .            .      2.2%        print('Action sequence:')
   274         1        0.1ms        0.1ms     30.1%        print(*actionSequence, sep='\n')
   275
   276         1            .            .      1.6%        print('\nState sequence:')
   277         1        0.1ms        0.1ms     36.6%        print(*stateSequence, sep='\n')
   278
   279         1            .            .      1.6%        print('\nPath cost:', pathCost)
```