```
Total time: 0.062s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.py
Function: expand at line 24

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    24                                                  @cpu
    25                                                  def expand(board):
    26      1748        0.5ms            .      0.9%        for i in range(len(board.data)):                                # to find the loca
    27      4362        1.2ms            .      1.9%            for j in range(len(board.data[i])):
    28      3488        1.0ms            .      1.7%                if board.data[i][j] == '*':
    29       437        0.1ms            .      0.2%                    location = [i,j];
    30       437        0.1ms            .      0.1%                    break
    31
    32       437        0.1ms            .      0.2%        actions = []
    33      1691        6.3ms            .     10.3%        for move in possible_actions(constants.board, location):        # to find all poss
    34      1254       52.2ms            .     84.7%            actions.append([result(location, move, board.data) , move]) # prepare all poss
    35
    36       437        0.1ms            .      0.2%        return actions                                                  # After expanding
```

```
Total time: 0.003s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.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       437        0.1ms            .      3.4%        actions = ["RIGHT","LEFT","UP","DOWN"]
    41       437        0.1ms            .      3.8%        actionstopeform = []
    42
    43      2185        0.5ms            .     17.0%        for x in actions:
    44                                                              # for moving right
    45      1748        0.5ms            .     15.3%            if x == "RIGHT":
    46       437        0.1ms            .      4.7%                if location[1]+1 < len(board):
    47       318        0.1ms            .      4.7%                    actionstopeform.append([x,location[0],location[1]+1])
    48                                                              # for moving left
    49      1311        0.3ms            .     10.6%            elif x == "LEFT":
    50       437        0.1ms            .      4.3%                if location[1]-1 >= 0:
    51       310        0.1ms            .      4.6%                    actionstopeform.append([x,location[0],location[1]-1])
    52                                                              # for moving up
    53       874        0.2ms            .      7.1%            elif x == "UP":
    54       437        0.1ms            .      4.2%                if location[0]-1 >= 0:
    55       314        0.1ms            .      4.6%                    actionstopeform.append([x,location[0]-1,location[1]])
    56                                                              # for moving down
    57       437        0.1ms            .      3.9%            elif x == "DOWN":
    58       437        0.1ms            .      4.4%                if location[0]+1 < len(board):
    59       312        0.1ms            .      4.4%                    actionstopeform.append([x,location[0]+1,location[1]])
    60
    61       437        0.1ms            .      3.1%        return actionstopeform
```

```
Total time: 0.049s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.py
Function: result at line 63

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    63                                                  @cpu
    64                                                  def result(location,action,board):
    65      1254       40.2ms            .     81.6%        newBoard = copy.deepcopy(board)                             # copy of a board so t
    66      1254        3.0ms            .      6.2%        temp = copy.deepcopy(newBoard[action[1]][action[2]])
    67      1254        2.9ms            .      6.0%        newBoard[action[1]][action[2]] = copy.deepcopy('*')
    68      1254        2.8ms            .      5.8%        newBoard[location[0]][location[1]]  = copy.deepcopy(temp)
    69      1254        0.2ms            .      0.5%        return newBoard                                             # return new board aft
```

```
Total time: 0.002s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.py
Function: hasCycle at line 73

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    73                                                  @cpu
    74                                                  def hasCycle(list):                                             # function to check if
    75       437        0.1ms            .      5.0%        s = set()
    76       437        0.1ms            .      3.6%        temp = list
    77      2512        0.5ms            .     21.7%        while (temp):
    78      2075        0.5ms            .     20.0%            if (temp in s):
    79                                                                  return True
    80      2075        0.6ms            .     25.1%            s.add(temp)
    81      2075        0.5ms            .     20.6%            temp = temp.prev
    82       437        0.1ms            .      4.0%        return False
```

```
Total time: 0.087s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.py
Function: idfs at line 86

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    86                                                  @memory_profiler.profile
    87                                                  @cpu
    88                                                  def idfs(board,depth):                              # function ITERATIVE-DEEPENING-SEA
    89         6            .            .         .        for step in range(depth):                       # for depth = 0 to ∞ do
    90         6       87.2ms       14.5ms    100.0%            result = depthFirstSearch(board, step)      # result ← DEPTH-LIMITED-SEARCH(pr
    91         6            .            .         .            if(result != cut_off):                      # if result 6= cutoff
    92         1            .            .         .                return result                           # then return result


Total time: 0.083s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.py
Function: depthFirstSearch at line 95

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
    95                                                  @cpu
    96
    97                                                  def depthFirstSearch(board, step):                  # function DEPTH-LIMITED-SEARCH(pr
    98         6        0.1ms            .      0.1%        frontier = queue.LifoQueue()                    # frontier ← a LIFO queue (stack)
    99         6            .            .         .        result = failure                                # result ← failure
   100         6            .            .         .        node = Node(data=board)
   101         6            .            .         .        frontier.put(node)
   102         6            .            .         .        maxQueueSize =1                                 # only for debug
   103
   104      1258        1.3ms            .      1.5%        while not frontier.empty():                     # while not IS-EMPTY(frontier ) do
   105      1253        4.4ms            .      5.3%            val = frontier.get()                        # node ← POP(frontier )
   106      1253        0.5ms            .      0.6%            if goalBoard == val.data:                   # if problem.IS-GOAL(node.STATE) t
   107         1            .            .         .                return val                              # return node
   108      1252        0.4ms            .      0.5%            if  val.depth > step:                       # if DEPTH(node) > ℓ then
   109       815        0.3ms            .      0.3%                result = cut_off                        # result ← cutoff
   110       437        4.9ms            .      5.8%            elif not hasCycle(val):                     # else if not IS-CYCLE(node) do
   111      1691       65.3ms            .     78.4%                for child in expand(val):               # for each child in EXPAND(problem
   112      1254        1.5ms            .      1.8%                    temp =  Node(data=child[0], depth =val.depth + 1 ,move= child[1] , pre
   113      1254        4.2ms            .      5.0%                    frontier.put(temp)                  # add child to frontier
   114      1254        0.4ms            .      0.5%                    maxQueueSize+=1                     # only for debug
   115
   116                                                      #print('Max queue size:', maxQueueSize)         # only for debug
   117         5            .            .         .        return result                                   # return result

Total time: 0.000s
File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/assignment_1/iterativeDeepeningSearch.py
Function: printStatistics at line 119

Line #      Hits         Time      Per Hit    % Time    Line Contents
=====================================================================
   119                                                  @cpu
   120                                                  def printStatistics(solution):
   121         1            .            .      2.0%        pathCost = 0
   122         1            .            .         .        stateSequence = []
   123         1            .            .         .        actionSequence = []
   124
   125         7            .            .      8.2%        while solution.prev != None:
   126         6            .            .      4.1%            stateSequence.insert(0, solution.data)
   127         6            .            .      6.1%            actionSequence.insert(0, solution.move)
   128         6            .            .      4.1%            solution = solution.prev
   129         6            .            .         .            pathCost += 1
   130
   131         1            .            .     12.2%        print('Action sequence:')
   132         1            .            .     26.5%        print(*actionSequence, sep='\n')
   133
   134         1            .            .      6.1%        print('\nState sequence:')
   135         1            .            .     24.5%        print(*stateSequence, sep='\n')
   136
   137         1            .            .      6.1%        print('\nPath cost:', pathCost)
```