

Total time: 0.061s

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	1748	0.5ms	.	0.7%	for i in range(len(board.data)):
64	4362	1.1ms	.	1.8%	for j in range(len(board.data[i])):
65	3488	1.0ms	.	1.6%	if board.data[i][j] == '*':
66	437	0.1ms	.	0.2%	location = [i,j];
67	437	0.1ms	.	0.1%	break
68					
69	437	0.1ms	.	0.2%	actions = []
70	1691	6.0ms	.	9.9%	for move in possible_actions(constants.board, location):
71	1254	51.6ms	.	85.3%	actions.append([result(location, move, board.data) , move])
72					
73	437	0.1ms	.	0.2%	return actions

Total time: 0.003s

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	437	0.1ms	.	4.4%	actions = ["RIGHT", "LEFT", "UP", "DOWN"]
78	437	0.1ms	.	3.5%	actionstopeform = []
79					
80	2185	0.5ms	.	18.0%	for x in actions:
81					# for moving right
82	1748	0.4ms	.	14.0%	if x == "RIGHT":
83	437	0.2ms	.	5.6%	if location[1]+1 < len(board):
84	318	0.1ms	.	3.9%	actionstopeform.append([x,location[0],location[1]+1])
85					# for moving left
86	1311	0.3ms	.	11.0%	elif x == "LEFT":
87	437	0.1ms	.	4.0%	if location[1]-1 >= 0:
88	310	0.1ms	.	4.4%	actionstopeform.append([x,location[0],location[1]-1])
89					# for moving up
90	874	0.2ms	.	6.7%	elif x == "UP":
91	437	0.1ms	.	3.7%	if location[0]-1 >= 0:
92	314	0.1ms	.	4.6%	actionstopeform.append([x,location[0]-1,location[1]])
93					# for moving down
94	437	0.1ms	.	3.7%	elif x == "DOWN":
95	437	0.1ms	.	4.9%	if location[0]+1 < len(board):
96	312	0.1ms	.	4.2%	actionstopeform.append([x,location[0]+1,location[1]])
97					
98	437	0.1ms	.	3.3%	return actionstopeform

Total time: 0.049s

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	1254	39.8ms	.	81.5%	newBoard = copy.deepcopy(board)
104	1254	3.0ms	.	6.1%	temp = copy.deepcopy(newBoard[action[1]][action[2]])
105	1254	2.9ms	.	6.0%	newBoard[action[1]][action[2]] = copy.deepcopy('*')
106	1254	2.8ms	.	5.8%	newBoard[location[0]][location[1]] = copy.deepcopy(temp)
107					# return new board after moving * - NIL to the new location
108	1254	0.3ms	.	0.6%	return newBoard

Total time: 0.002s

File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py

Function: hasCycle at line 130

Line #	Hits	Time	Per Hit	% Time	Line Contents
=====					
130					@cpu
131					def hasCycle(list):
132	437	0.1ms	.	5.4%	s = set()
133	437	0.1ms	.	3.9%	temp = list
134	2512	0.5ms	.	22.0%	while (temp):
135	2075	0.4ms	.	20.1%	if (temp in s):
136					return True
137	2075	0.5ms	.	24.7%	s.add(temp)
138	2075	0.5ms	.	20.7%	temp = temp.prev
139	437	0.1ms	.	3.1%	return False

Total time: 0.086s

File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py

Function: idfs at line 143

Line #	Hits	Time	Per Hit	% Time	Line Contents
=====					
143					@cpu
144					def idfs(board,depth):
145	6	.	.	.	for step in range(depth):
146	6	85.6ms	14.3ms	100.0%	result = depthFirstSearch(board, step)

```

147         6         .         .         .         if(result != cut_off):
148         1         .         .         .         return result

```

Total time: 0.082s

File: /Users/rishabhjain/Documents/Masters/SEM 2/Aritificial Intelligence/Program/Program 1/program_1.py

Function: depthFirstSearch at line 150

Line #	Hits	Time	Per Hit	% Time	Line Contents
=====					
150					@cpu
151					def depthFirstSearch(board, step):
152	6	.	.	.	result = failure
153	6	0.1ms	.	0.1%	frontier = queue.LifoQueue()
154	6	.	.	.	node = Node(data=board)
155	6	.	.	.	frontier.put(node)
156	6	.	.	.	maxQueueSize =1
157					
158	1258	1.2ms	.	1.5%	while not frontier.empty():
159	1253	4.3ms	.	5.3%	val = frontier.get()
160	1253	0.5ms	.	0.6%	if goalBoard == val.data:
161	1	.	.	.	return val
162	1252	0.4ms	.	0.5%	if val.depth > step:
163	815	0.3ms	.	0.3%	result = cut_off
164	437	4.6ms	.	5.7%	elif not hasCycle(val):
165	1691	64.2ms	.	78.5%	for child in expand(val):
166	1254	1.5ms	.	1.8%	temp = Node(data=child[0], depth =val.depth + 1 ,move= child[1] , pre
167	1254	4.0ms	.	4.9%	frontier.put(temp)
168	1254	0.4ms	.	0.5%	maxQueueSize+=1
169					
170	5	0.1ms	.	0.1%	print('Max queue size:', maxQueueSize)
171	5	.	.	.	return result