

Total time: 12.469s

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	335432	150.6ms	.	1.2%	for i in range(len(board.data)):
51	838579	247.1ms	.	2.0%	for j in range(len(board.data[i])):
52	670863	209.4ms	.	1.7%	if board.data[i][j] == '*':
53	83858	25.4ms	.	0.2%	location = [i,j];
54	83858	18.6ms	.	0.1%	break
55					
56	83858	18.7ms	.	0.2%	actions = []
57	308279	1.39s	.	11.1%	for move in possible_actions(constants.board, location):
58	224421	10.39s	.	83.3%	actions.append([result(location, move, board.data) , move])
59					
60	83858	20.2ms	.	0.2%	return actions

Total time: 0.647s

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	83858	29.6ms	.	4.6%	actions = ["RIGHT", "LEFT", "UP", "DOWN"]
65	83858	22.5ms	.	3.5%	actionstopeform = []
66					
67	419290	116.4ms	.	18.0%	for x in actions:
68					# for moving right
69	335432	90.3ms	.	14.0%	if x == "RIGHT":
70	83858	35.9ms	.	5.6%	if location[1]+1 < len(board):
71	56129	30.5ms	.	4.7%	actionstopeform.append([x,location[0],location[1]+1])
72					# for moving left
73	251574	70.9ms	.	11.0%	elif x == "LEFT":
74	83858	28.8ms	.	4.4%	if location[1]-1 >= 0:
75	56128	28.0ms	.	4.3%	actionstopeform.append([x,location[0],location[1]-1])
76					# for moving up
77	167716	44.5ms	.	6.9%	elif x == "UP":
78	83858	28.1ms	.	4.3%	if location[0]-1 >= 0:
79	56070	24.2ms	.	3.7%	actionstopeform.append([x,location[0]-1,location[1]])
80					# for moving down
81	83858	22.1ms	.	3.4%	elif x == "DOWN":
82	83858	29.8ms	.	4.6%	if location[0]+1 < len(board):
83	56094	24.6ms	.	3.8%	actionstopeform.append([x,location[0]+1,location[1]])
84					
85	83858	20.4ms	.	3.2%	return actionstopeform

Total time: 9.821s

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	224421	8.11s	.	82.5%	newBoard = copy.deepcopy(board)
91	224421	571.3ms	.	5.8%	temp = copy.deepcopy(newBoard[action[1]][action[2]])
92	224421	554.3ms	.	5.6%	newBoard[action[1]][action[2]] = copy.deepcopy('*')
93	224421	538.0ms	.	5.5%	newBoard[location[0]][location[1]] = copy.deepcopy(temp)
94					# return new board after moving * - NIL to the new location
95	224421	50.3ms	.	0.5%	return newBoard

Total time: 379.948s

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

Function: bfs at line 139

Line #	Hits	Time	Per Hit	% Time	Line Contents
=====					
139					@cpu
140					def bfs(board):
141	1	.	.	.	frontier = queue.Queue()
142	1	.	.	.	node = Node(data = board)
143	1	.	.	.	frontier.put(node)
144	1	.	.	.	maxQueueSize = 1
145	1	.	.	.	if constants.goalBoard == node.data:
146					print('mill gya:', maxQueueSize)
147					return node
148					
149	1	.	.	.	reached = []
150	1	.	.	.	reached.append(board)
151					
152	83858	276.9ms	.	0.1%	while not frontier.empty():
153	83858	545.4ms	.	0.1%	val = frontier.get()
154	308277	13.45s	.	3.5%	for child in expand(val):
155	224420	605.0ms	.	0.2%	s = Node(data=child[0], depth = val.depth + 1, move= child[1] , prev=val)
156					
157	224420	143.0ms	.	.	if goalBoard == s.data:
158	1	0.1ms	0.1ms	.	print('Max queue size:', maxQueueSize)
159	1	.	.	.	return s
160	224419	363.60s	1.6ms	95.7%	if s.data not in reached:

```

161 105271 158.3ms . . reached.append(s.data)
162 105271 1.12s . 0.3% frontier.put(s)
163 105271 44.7ms . . maxQueueSize+=1
164
165 print('Max queue size:', maxQueueSize)
166
167 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 258

Line #	Hits	Time	Per Hit	% Time	Line Contents
258					@cpu
259					def printStatistics(solution):
260	1	.	.	0.3%	pathCost = 0
261	1	.	.	.	stateSequence = []
262	1	.	.	.	actionSequence = []
263					
264	24	.	.	3.8%	while solution.prev != None:
265	23	.	.	4.5%	stateSequence.insert(0, solution.data)
266	23	.	.	1.7%	actionSequence.insert(0, solution.move)
267	23	.	.	3.8%	solution = solution.prev
268	23	.	.	1.7%	pathCost += 1
269					
270	1	.	.	6.6%	print('Action sequence:')
271	1	0.1ms	0.1ms	24.7%	print(*actionSequence, sep='\n')
272					
273	1	.	.	1.0%	print('\nState sequence:')
274	1	0.1ms	0.1ms	49.5%	print(*stateSequence, sep='\n')
275					
276	1	.	.	2.1%	print('\nPath cost:', pathCost)