



Tecnológico de Monterrey

Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Engineering in Computational Technologies
Analysis and Design of Advanced Algorithms

Homework 14: 8 Tree

Group: 607
Team #3

Luis Salomón Flores Ugalde

Santiago Quintana Moreno A01571222
Miguel Ángel Álvarez Hermida a01722925

File Edit Selection View Go Run Terminal Help

A01571222_A01722925_8puzzle.py U X

Homework14_Puzzle > A01571222_A01722925_8puzzle.py > Node > _init_

```
1 # Analysis and Design of Advanced Algorithms
2 # Group #607
3 # Team 3
4 # Luis Salomón Flores Ugalde
5
6 # Santiago Quintana Moreno A01571222
7 # Miguel Ángel Álvarez Hermida A01722925
8
9 # ----- HOMEWORK 14 PUZZLE SOLVER -----
10
11 import random
12 import itertools
13 import collections
14 import time
15
16 class Node:
17     """
18     A class representing an Solver node
19     - 'puzzle' is a Puzzle instance see that move() creates a new puzzle depending on the move
20     - 'parent' is the preceding node generated by the solver, if any
21     - 'action' is the action taken to produce puzzle, if any
22     """
23
24     def __init__(self, puzzle, parent=None, action=None):
25         self.puzzle = puzzle
26         self.parent = parent
27         self.action = action
28         if parent:
29             self.g = parent.g + 1
30         else:
31             self.g = 0
32
33     def state(self):
34         """
35         Return a hashable representation of self
36         """
37         return str(self)
38
39     def path(self):
40         """
41         Reconstruct path by walking back from current node to start
42         """
43         node, p = self, []
44         while node:
45             p.append(node)
46             node = node.parent
47         return p[::-1]
```

powershell X

```
(. .\jupy) PS D:\1-SQM\1-UNIVERSIDAD\5- QUINTO SEMESTRE\2-Advanced Algorithms> & "D:\1-SQM\1-UNIVERSIDAD\5- QUINTO SEMESTRE\2-Advanced Algorithms\jupyScripts\python.exe" "d:\1-SQM\1-UNIVERSIDAD\5- QUINTO SEMESTRE\2-Advanced Algorithms\Homework14_Puzzle\A01571222_A01722925_8puzzle.py"
None
[1, 2, 3]
[4, 5, 0]
[6, 7, 8]

U
[1, 2, 0]
[4, 5, 3]
[6, 7, 8]

L
[1, 0, 2]
[4, 5, 3]
[6, 7, 8]

D
[1, 5, 2]
[4, 0, 3]
[6, 7, 8]

L
[1, 5, 2]
[0, 4, 3]
[6, 7, 8]

U
[0, 5, 2]
[1, 4, 3]
[6, 7, 8]

R
[5, 0, 2]
[1, 4, 3]
[6, 7, 8]

R
[5, 2, 0]
[1, 4, 3]
[6, 7, 8]

D
[5, 2, 3]
[1, 4, 0]
[6, 7, 8]

D
[5, 2, 3]
[1, 4, 8]
[6, 7, 0]

L
[5, 2, 3]
```

main* AplicativoBodega/AplicativoApp#12 needs reviewers 0 0 0 Go Live

File Edit Selection View Go Run Terminal Help

A01571222_A01722925_8puzzle.py U X

Homework14_Puzzle > A01571222_A01722925_8puzzle.py > Node > _init_

```
16 class Node:
40     def path(self):
48         return p[::-1]
49
50     def solved(self):
51         """ Wrapper to check if 'puzzle' is solved """
52         return self.puzzle.solved()
53
54     def actions(self):
55         """ Wrapper for 'actions' accessible at current state """
56         return self.puzzle.actions()
57
58     def h(self):
59         """ Calculate h value using Manhattan distance heuristic """
60         return self.puzzle.manhattan()
61
62     def f(self):
63         """
64         Return f(n) = g(n) + h(n) for A*.
65         """
66         return self.g + self.h()
67
68     def __str__(self):
69         return str(self.puzzle)
70
71
72 class Solver:
73     """
74     '8-puzzle' solver
75     - 'start' is a Puzzle instance
76     """
77
78     def __init__(self, start):
79         self.start = start
80
81     def solve(self):
82         queue = collections.deque([Node(self.start)])
83         seen = set()
84         seen.add(queue[0].state())
85
86         while queue:
87             queue = collections.deque(sorted(queue, key=lambda node: node.f()))
88             node = queue.popleft()
89
90             if node.solved():
91                 return node.path()
92
93             for move, action in node.actions():
```

powershell X

(.jupyter) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> 8 "D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\jupyter\Scripts\python.exe" "d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\Homework14_Puzzle\A01571222_A01722925_8puzzle.py"

None
[1, 2, 3]
[4, 5, 0]
[6, 7, 8]

U
[1, 2, 0]
[4, 5, 3]
[6, 7, 8]

L
[1, 0, 2]
[4, 5, 3]
[6, 7, 8]

D
[1, 5, 2]
[4, 0, 3]
[6, 7, 8]

L
[1, 5, 2]
[0, 4, 3]
[6, 7, 8]

U
[0, 5, 2]
[1, 4, 3]
[6, 7, 8]

R
[5, 0, 2]
[1, 4, 3]
[6, 7, 8]

R
[5, 2, 0]
[1, 4, 3]
[6, 7, 8]

D
[5, 2, 3]
[1, 4, 0]
[6, 7, 8]

D
[5, 2, 3]
[1, 4, 8]
[6, 7, 0]

L
[5, 2, 3]

main* AplicativoBodega/AplicativoApp#12 needs reviews 0 0 0

File Edit Selection View Go Run Terminal Help

A01571222_A01722925_8puzzle.py U X

Homework14_Puzzle > A01571222_A01722925_8puzzle.py > Node > _init_

```
72 class Solver:
81     def solve(self):
82
83         for move, action in node.actions():
84             child = Node(move(), node, action)
85
86             if child.state() not in seen:
87                 queue.append(child)
88                 seen.add(child.state())
89
90
91 class Puzzle:
92     """
93     A class representing an '8-puzzle'.
94     - 'board' should be a square list of lists with integer entries 0...width^2 - 1
95     | e.g. [[1,2,3],[4,0,6],[7,5,8]]
96     - 'goal_state' is another board in the same format.
97     """
98
99     def __init__(self, board, goal_state=None):
100         self.width = len(board[0])
101         self.board = board
102         self.goal = goal_state
103
104         # Default goal state
105         if goal_state is None:
106             self.goal = [[1, 2, 3],
107                          [4, 5, 6],
108                          [7, 8, 0]]
109         else:
110             self.goal = goal_state
111
112         # Precompute goal positions for Manhattan
113         self.goal_pos = {}
114         for i in range(self.width):
115             for j in range(self.width):
116                 v = self.goal[i][j]
117                 self.goal_pos[v] = (i, j)
118
119     def solved(self):
120         """
121         The puzzle is solved if the current board equals the goal board.
122         """
123         return self.board == self.goal
124
125     def actions(self):
126         """
127         Return a list of 'move', 'action' pairs. 'move' can be called
128         to return a new puzzle that results in sliding the '0' tile in
129         the direction of 'action'.
```

...

powershell X

```
(. .\jupyter) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & "D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\jupyter\Scripts\python.exe" "d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\Homework14_Puzzle\A01571222_A01722925_8puzzle.py"
None
[1, 2, 3]
[4, 5, 0]
[6, 7, 8]
U
[1, 2, 0]
[4, 5, 3]
[6, 7, 8]
L
[1, 0, 2]
[4, 5, 3]
[6, 7, 8]
D
[1, 5, 2]
[4, 0, 3]
[6, 7, 8]
L
[1, 5, 2]
[0, 4, 3]
[6, 7, 8]
U
[0, 5, 2]
[1, 4, 3]
[6, 7, 8]
R
[5, 0, 2]
[1, 4, 3]
[6, 7, 8]
R
[5, 2, 0]
[1, 4, 3]
[6, 7, 8]
D
[5, 2, 3]
[1, 4, 0]
[6, 7, 8]
D
[5, 2, 3]
[1, 4, 8]
[6, 7, 0]
L
[5, 2, 3]
```

main

AplicativoBodega/AplicativoApp#12 needs reviews

0 0

File Edit Selection View Go Run Terminal Help

A01571222_A01722925_8puzzle.py U X

Homework14_Puzzle > A01571222_A01722925_8puzzle.py > Node > __init__

```
101 class Puzzle:
102     def actions(self):
103         """
104         to return a new puzzle that results in sliding the '0' tile in
105         the direction of 'action'.
106         """
107
108     def create_move(at, to):
109         return lambda: self._move(at, to)
110
111     moves = []
112     for i, j in itertools.product(range(self.width),
113                                   range(self.height)):
114         dirs = {'R': (i, j - 1),
115                'L': (i, j + 1),
116                'D': (i - 1, j),
117                'U': (i + 1, j)}
118
119         for action, (r, c) in dirs.items():
120             if 0 <= r < self.width and 0 <= c < self.height and self.board[r][c] == 0:
121                 move = create_move((i, j), (r, c), action)
122                 moves.append(move)
123
124     return moves
125
126 def manhattan(self):
127     """
128     Manhattan distance to the current goal_state.
129     Uses goal_pos computed from self.goal.
130     """
131
132     distance = 0
133     for i in range(self.width):
134         for j in range(self.height):
135             v = self.board[i][j]
136             if v == 0:
137                 continue # usually ignore blank
138             gi, gj = self.goal_pos[v]
139             distance += abs(gi - i) + abs(gj - j)
140
141     return distance
142
143 def copy(self):
144     """
145     Return a new puzzle with the same board as 'self'
146     """
147
148     board = []
149     for row in self.board:
150         board.append([x for x in row])
151     return Puzzle(board, goal_state=self.goal)
```

powershell X

```
(. .\jupyter) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & "D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\jupyter\Scripts\python.exe" "d:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\Homework14_Puzzle\A01571222_A01722925_8puzzle.py"
None
[1, 2, 3]
[4, 5, 0]
[6, 7, 8]

U
[1, 2, 0]
[4, 5, 3]
[6, 7, 8]

L
[1, 0, 2]
[4, 5, 3]
[6, 7, 8]

D
[1, 5, 2]
[4, 0, 3]
[6, 7, 8]

L
[1, 5, 2]
[0, 4, 3]
[6, 7, 8]

U
[0, 5, 2]
[1, 4, 3]
[6, 7, 8]

R
[5, 0, 2]
[1, 4, 3]
[6, 7, 8]

R
[5, 2, 0]
[1, 4, 3]
[6, 7, 8]

D
[5, 2, 3]
[1, 4, 0]
[6, 7, 8]

D
[5, 2, 3]
[1, 4, 8]
[6, 7, 0]

L
[5, 2, 3]
```

main

AplicativoBodega/AplicativoApp#12 needs reviews

0 0 Live

... powershell X

[illegible]

File Edit Selection View Go Run Terminal Help

A01571222_A01722925_8puzzle.py U x

101 class Puzzle:

181 return Puzzle(board, goal_state=self.goal)

182

183

184

185 def move(self, at, to):

186 """

187 Return a new puzzle where 'at' and 'to' tiles have been swapped.

188 NOTE: all moves should be 'actions' that have been executed

189 """

190 copy = self.copy()

191 i, j = at

192 r, c = to

193 copy.board[i][j], copy.board[r][c] = copy.board[r][c], copy.board[i][j]

194 return copy

195

196 def pprint(self):

197 for row in self.board:

198 print(row)

199 print()

200

201 def __str__(self):

202 return ''.join(map(str, self))

203

204 def __iter__(self):

205 for row in self.board:

206 yield from row

207

208 board = [[1, 2, 3], [4, 5, 0], [6, 7, 8]]

209 goal_state = [[5, 7, 3], [1, 0, 8], [6, 2, 4]]

210 puzzle = Puzzle(board, goal_state)

211

212 s = Solver(puzzle)

213 tic = time.perf_counter()

214 p = s.solve()

215 toc = time.perf_counter()

216

217 steps = 0

218 for node in p:

219 print(node.action)

220 node.puzzle.pprint()

221 steps += 1

222

223 print("Total number of steps: " + str(steps))

224 print("Total amount of time in search: " + str(toc - tic) + " second(s)")

225

...

powershell X

(.jupyter) PS D:\1-SQM\1-UNIVERSIDAD\5- QUINTO SEMESTRE\2-Advanced Algorithms> 8 "D:\1-SQM\1-UNIVERSIDAD\5- QUINTO SEMESTRE\2-Advanced Algorithms\jupyter\Scripts\python.exe" "d:\1-SQM\1-UNIVERSIDAD\5- QUINTO SEMESTRE\2-Advanced Algorithms\Homework14_Puzzle\A01571222_A01722925_8puzzle.py"

None

[1, 2, 3]

[4, 5, 0]

[6, 7, 8]

U

[1, 2, 0]

[4, 5, 3]

[6, 7, 8]

L

[1, 0, 2]

[4, 5, 3]

[6, 7, 8]

D

[1, 5, 2]

[4, 0, 3]

[6, 7, 8]

L

[1, 5, 2]

[0, 4, 3]

[6, 7, 8]

U

[0, 5, 2]

[1, 4, 3]

[6, 7, 8]

R

[5, 0, 2]

[1, 4, 3]

[6, 7, 8]

R

[5, 2, 0]

[1, 4, 3]

[6, 7, 8]

D

[5, 2, 3]

[1, 4, 0]

[6, 7, 8]

D

[5, 2, 3]

[1, 4, 8]

[6, 7, 0]

L

[5, 2, 3]

main*

AplicativoBodega/AplicativoApp#12 needs reviews

0 0 0

```
A01571222_A01722925_8puzzle.py U X
Homework14_Puzzle > A01571222_A01722925_8puzzle.py > Node > _init_
181 class Puzzle:
182
183     return Puzzle(board, goal_state=self.goal)
184
185 def _move(self, at, to):
186     """
187     Return a new puzzle where 'at' and 'to' tiles have been swapped.
188     NOTE: all moves should be 'actions' that have been executed
189     """
190     copy = self.copy()
191     i, j = at
192     r, c = to
193     copy.board[i][j], copy.board[r][c] = copy.board[r][c], copy.board[i][j]
194     return copy
195
196 def pprint(self):
197     for row in self.board:
198         print(row)
199     print()
200
201 def __str__(self):
202     return ''.join(map(str, self))
203
204 def __iter__(self):
205     for row in self.board:
206         yield from row
207
208 board = [[1, 2, 3], [4, 5, 0], [6, 7, 8]]
209 goal_state = [[5, 7, 3], [1, 0, 8], [6, 2, 4]]
210 puzzle = Puzzle(board, goal_state)
211
212 s = Solver(puzzle)
213 tic = time.perf_counter()
214 p = s.solve()
215 toc = time.perf_counter()
216
217 steps = 0
218 for node in p:
219     print(node.action)
220     node.puzzle.pprint()
221     steps += 1
222
223 print("Total number of steps: " + str(steps))
224 print("Total amount of time in search: " + str(toc - tic) + " second(s)")
225
```

powershell X

```
L
[5, 2, 3]
[1, 4, 8]
[6, 0, 7]

U
[5, 2, 3]
[1, 0, 8]
[6, 4, 7]

U
[5, 0, 3]
[1, 2, 8]
[6, 4, 7]

R
[5, 3, 0]
[1, 2, 8]
[6, 4, 7]

D
[5, 3, 8]
[1, 2, 0]
[6, 4, 7]

D
[5, 3, 8]
[1, 2, 7]
[6, 4, 0]

L
[5, 3, 8]
[1, 2, 7]
[6, 0, 4]

U
[5, 3, 8]
[1, 0, 7]
[6, 2, 4]

R
[5, 3, 8]
[1, 7, 0]
[6, 2, 4]

U
[5, 3, 0]
[1, 7, 8]
[6, 2, 4]

L
[5, 0, 3]
[1, 7, 8]
[6, 2, 4]
```


File Edit Selection View Go Run Terminal Help

A01571222_A01722925_8puzzle.py U X

Homework14_Puzzle > A01571222_A01722925_8puzzle.py > Node > _init_

181 class Puzzle:

181 return Puzzle(board, goal_state=self.goal)

182

183 def move(self, at, to):

184

185 Return a new puzzle where 'at' and 'to' tiles have been swapped.

186 NOTE: all moves should be 'actions' that have been executed

187

188 copy = self.copy()

189 i, j = at

190 r, c = to

191 copy.board[i][j], copy.board[r][c] = copy.board[r][c], copy.board[i][j]

192 return copy

193

194 def pprint(self):

195 for row in self.board:

196 print(row)

197 print()

198

199 def __str__(self):

200 return ''.join(map(str, self))

201

202 def __iter__(self):

203 for row in self.board:

204 yield from row

205

206

207 board = [[1, 2, 3], [4, 5, 0], [6, 7, 8]]

208 goal_state = [[5, 7, 3], [1, 0, 8], [6, 2, 4]]

209 puzzle = Puzzle(board, goal_state)

210

211 s = Solver(puzzle)

212 tic = time.perf_counter()

213 p = s.solve()

214 toc = time.perf_counter()

215

216 steps = 0

217 for node in p:

218 print(node.action)

219 node.puzzle.pprint()

220 steps += 1

221

222 print("Total number of steps: " + str(steps))

223 print("Total amount of time in search: " + str(toc - tic) + " second(s)")

224

225

powershell X

[6, 4, 7]

U

[5, 0, 3]

[1, 2, 8]

[6, 4, 7]

R

[5, 3, 0]

[1, 2, 8]

[6, 4, 7]

D

[5, 3, 8]

[1, 2, 0]

[6, 4, 7]

D

[5, 3, 8]

[1, 2, 7]

[6, 4, 0]

L

[5, 3, 8]

[1, 2, 7]

[6, 0, 4]

U

[5, 3, 8]

[1, 0, 7]

[6, 2, 4]

R

[5, 3, 8]

[1, 7, 0]

[6, 2, 4]

U

[5, 3, 0]

[1, 7, 8]

[6, 2, 4]

L

[5, 0, 3]

[1, 7, 8]

[6, 2, 4]

D

[5, 7, 3]

[1, 0, 8]

[6, 2, 4]

Total number of steps: 22

Total amount of time in search: 0.2641739000027883 second(s)

(.jupy) PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>

main*

AplicativoBodega/AplicativoApp#12 needs reviews

0 0 0

Go Live

https://colab.research.google.com/drive/14YoHYCqee8-XPWc_uEUSPSjNXZ6M_Oa5?usp=sharing