

ENPM 661

Project-1 Report

Submitted by : Aaradhana Kannan Subramanian

Language: Python

File name: 8_puzzle_final.py

Modules to be imported: numpy, os

Working:

The implementation of 8-puzzle works as follows:

1. It checks for a valid input. It provides restrictions for user entry such that the same number cannot be entered twice and limits the range of entry from 0 to 8.
2. During initial checks, the program also checks to see if the given sequence is actually solvable. If it is not a valid sequence, the program exits.
3. From the user-defined input, the first node is created. It is then sent to the `node_creation()` function.
4. The `node_creation()` function initializes the lists `node_q`, `seen` and `visited`. The list `node_q` acts as a queue. It appends the created nodes and pops them out one by one as and when their child nodes are created. Breadth first search is implemented to create the nodes.
5. Starting with the root node, a for loop iterates through the possible actions that can be performed (up, down, left, right). For each action that is possible, a new node is created with root node as its parent. The child node is then appended to the queue.
6. When the list element is popped, the root node is exited from the queue. The created nodes are appended to the lists `seen` and `visited` as well.
7. The next iteration performs the actions on the first child node and so on. If a child node's data is already present in the list `seen`, it is not appended and therefore is destroyed in the next iteration.
8. Every iteration checks if the child node created has the same data as goal node. If it is, it breaks from the function and returns the current node, and list of `seen` and `visited` nodes.
9. The `trackback()` function tracks the parent of the child and its parent and so on until it hits the node which has no parent i.e. the root node. These nodes are saved in a list.
10. The `writeToFile ()` function creates the three required files with the path information, node information and the list of explored nodes.
11. The program also prints to the console as shown in the figures. It displays each node along with the node number and the move that was made to reach that node until it reaches the goal node.

12. Three output files are created. Nodes.txt – list of all node data created, nodeinfo.txt – node number, parent number and cost for nodes in the path to goal node and Nodepath.txt – data stored in nodes from root to goal node.

Output :

```
Enter the 1 number: 81
Number must be between 0 and 8
Enter the 1 number: 8
Enter the 2 number: 1
Enter the 3 number: 2
Enter the 4 number: 0
Enter the 5 number: 4
Enter the 6 number: 4
input repeated, please enter another number
Enter the 6 number: 3
Enter the 7 number: 7
Enter the 8 number: 6
Enter the 9 number: 5
Unsolvable Input
The program will now exit
```

Fig 1. Unsolvable input

```
Enter the 1 number: 1
Enter the 2 number: 2
Enter the 3 number: 3
Enter the 4 number: 4
Enter the 5 number: 0
Enter the 6 number: 5
Enter the 7 number: 6
Enter the 8 number: 7
Enter the 9 number: 8
Input is valid
node_creation running
printing
Move : None      Result : [[1 2 3]
[4 0 5]          node number:0
[6 7 8]]
Move : right     Result : [[1 2 3]
[4 5 0]          node number:4
[6 7 8]]
Move : down      Result : [[1 2 3]
[4 5 8]          node number:14
[6 7 0]]
Move : left      Result : [[1 2 3]
[4 5 8]          node number:30
[6 0 7]]
Move : left      Result : [[1 2 3]
[4 5 8]          node number:52
[0 6 7]]

Move : right     Result : [[1 2 3]
[5 6 8]          node number:519
[4 7 0]]
Move : up        Result : [[1 2 3]
[5 6 0]          node number:855
[4 7 8]]
Move : left      Result : [[1 2 3]
[5 0 6]          node number:1423
[4 7 8]]
Move : left      Result : [[1 2 3]
[0 5 6]          node number:2297
[4 7 8]]
Move : down      Result : [[1 2 3]
[4 5 6]          node number:3741
[0 7 8]]
Move : right     Result : [[1 2 3]
[4 5 6]          node number:6016
[7 0 8]]
Move : right     Result : [[1 2 3]
[4 5 6]          node number:9714
[7 8 0]]
```

Fig 2. Solved output