Christian Carreras

CSC 548

02/27/2017

Project #1: 26-Puzzle Description

This project was created by modifying the given 8-puzzle code and modifying it to work in the 3rd dimension, thus creating a 26-puzzle. Following are the steps used to find the shortest path.

1. Print the initial configuration given as the first argument
2. Initialize the waiting lists used for nodes, f-values, and IDs to an empty list
3. Find the ID of the goal configuration in order to check for a match
    a. A unique ID is given to every configuration by adding its values to a sum multiplied by ten every iteration through its levels, rows, and columns
4. Create a node from the initial configuration
    a. A node's first member is the ID of the initial configuration
    b. The second member is the initial configuration (a 3D array)
    c. The third member is the g-value (this will be zero to start)
    d. The fourth member is the Manhattan Distance between the initial configuration and the goal configuration
        i. The Manhattan Distance is calculated by finding the location of each number in both configurations and subtracting the distance by level, row, and column. The absolute value of the differences is added together for each number from one to twenty-six
    e. The fifth member is the f-value
        i. The f-value equals the sum of the g-value and the Manhattan Distance
    f. The sixth member is the current path to the node (this will be empty to start)
5. Create the node to process (current node) as the initial node created
6. Find all current moves that can be made from the current node
    a. Generate all the moves that can be made from the current configuration: up, down, north, south, east, or west

   b. Make a new move

      i. Adjust the position of the zero value by adding the tuple of the current move which will have a single one or negative one value coordinating with the move to be made.

      ii. Create a new configuration for the new move

   c. Create a new node from the new configuration created from the move

      i. The new Manhattan Distance will be between the new configuration and the goal configuration

   d. The path will be copied from the previous node's path with the addition to the move just made

   e. Create a list of all nodes neighboring the current node

7. Filter the neighbors created from finding all the moves that can be made

   a. Loop through all moves (nodes)

   b. Avoid all nodes seen before unless it is better

      i. If the g-value of the new node is less than old node replace it

   c. If the current node was not seen before or was better than an older duplicate node add it to all the waiting lists

8. The new current node will be the first node in the waiting list

9. All the waiting lists will now remove their first node

10. Loop back to #6 until the current node's ID matches the ID of the goal configuration

11. Return the path of the current node once the goal configuration was reached (i.e. the cumulative path up to this node)