Group Members:

1. Qui, Christian Michael
2. Regencia, Josiah Eleazar

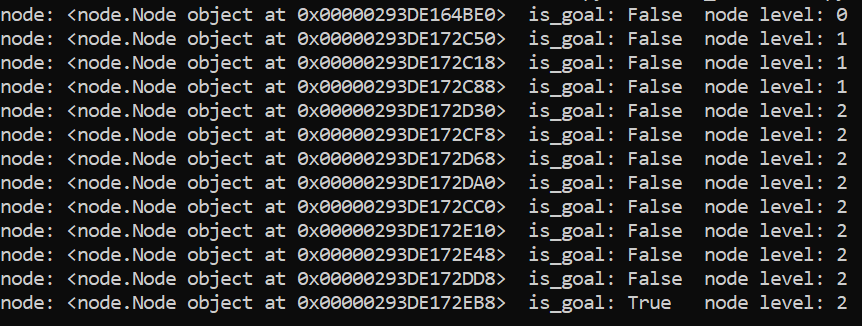
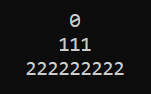
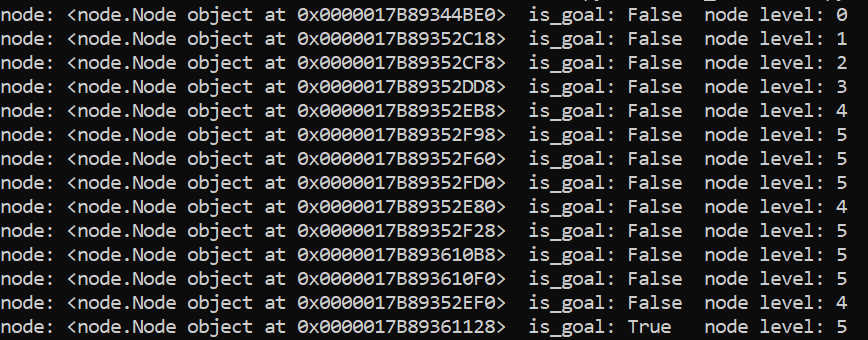
Requirements:

1. Python2.7.\* or Python3

Notes on how to run the homework:

1. Main file is tree\_traversal.py
2. Tree\_traversal.py accepts an argument to select whether the program will run a dfs or a bfs.
   1. To run bfs: python tree\_traversal.py bfs
   2. To run dfs: python tree\_traversal.py dfs
3. The arguments 'dfs' or 'bfs' are not case-sensitive. So it can be DFS or Dfs.

Notes on the output:

1. For each node that has been traversed, it will output a line until goal has been reached in the following format:
   * Node: <node address>, is\_goal: <Boolean>, node level: <int>
     + We decided to output the node address since there is no context yet. This program was made to be general as possible in preparation of the project.
2. Once goal has been reached (is\_goal: True), a visualization of the tree will be printed to show the path traversal. Implementation of this can be seen in tree\_view.py.
   * For best results, expand the cmd/terminal window to see the correct structure.
   * Example of BFS that traversed through 13 nodes:
   * 
   * 
   * Example of DFS that traversed 14 nodes:
   * 
   * 
     + \* Visualization of the DFS is a bit sparse due to the length of the 5th level, which is 243 nodes

Notes on the Implementation:

1. For this homework, we initially decided that the basis of a reached goal state is an integer generated randomly (See node class). If the basis is equal to 8, it will return True, which will cause the program to terminate. Otherwise, it will return False, and continue traversing.
2. In this implementation, each node has a 5% chance to be a goal state (1/20).
3. The children of a node are implemented as a list. This helps in making the branching factor dynamic.
   1. children[0] is implemented as the leftmost side of the list
   2. children[length-1] is implemented as the rightmost side of the list
4. DFS and BFS functions have the same implementation. The only differences are the following:
   1. Data structure that was used for each traversal type (Queue and Stack for BFS and DFS, respectively)
   2. The placement of the level tracker (Different algorithm for BFS and DFS)