Homework Assignment 2

Dalton Cole

September 15th, 2016

1 Algorithm Description

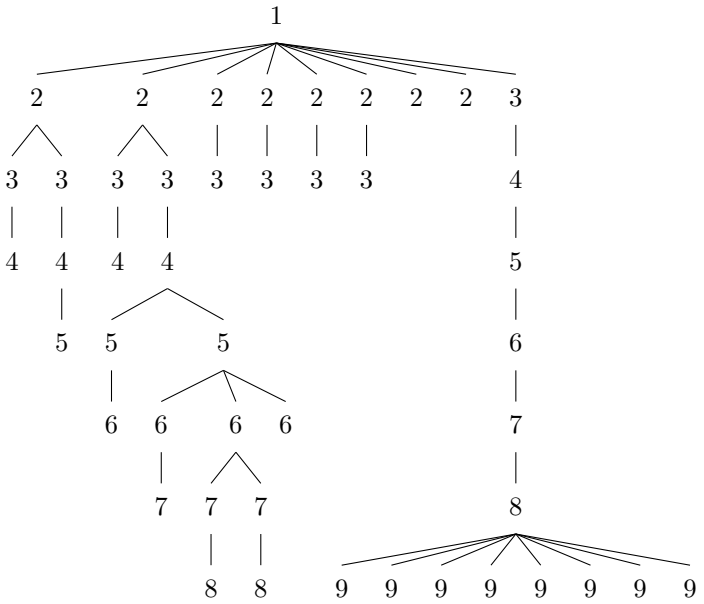
My algorithm for the Knight's Tour problem uses Depth First Search (DFS) using Warnsdorf's rule as a heuristic. Depth first search uses a stack that pushes possible moves onto the stack in order from lowest priority to highest. Priority is determined using Warnsdorf's rule. The next move is decided by how many moves the next knight could take. The next knight that can take the fewest future moves is added onto the stack last.

2 Pseudo Code

Algorithm 1 Pseudo Code

```
1:  procedure KNIGHT'S TOUR
2:      board = initial board
3:      stack.push(board)
4:      Solved = false
5:      while Solved = false do
6:          board ← stack.pop()
7:          if board is Solved then
8:              Quit Loop
9:          states ← board.moves()
10:         orderedStates ← states.order()
11:         stack.push(orderedStates)
```

3 Algorithm Illistraction



As illistrated by the tree above, the DFS algorithm creates a tree like stucture using a stack that adds up to 8 nodes per parent. The order in which the nodes are added depend on the number of children they will produce. If a child will have fewer children, it will be added before other children. By visiting states that are unlikely to be visited again first, we increase the odds of completing the tour.

4 Complexity

The worst possible case for this algorithm is $O(8^{n^2-1})$. This is because, for every possible node, 8 moves have to be checked. Since I am using DFS, $n^2 - 1$ nodes may be generated, where n is the size of the board.

$$T(n) \leq 8T(n^2 - 2)$$
$$T(n) \leq 8(8^{n^2-2})$$
$$T(n) \leq 8^{n^2-1}$$