Homework 1: The Knight's Tour

Kyle Hatfield kthvg5@mst.edu

September 14, 2016

1. The starting point for my algorithm was a Depth First Search (DFS). I designed a node data structure which contained a bool to note whether or not that node had been visited and a set of tuples representing the coordinates of the node's neighbors. A dictionary was used to store each of these nodes with a tuple containing their respective coordinates used at the key.

   Where this was efficient enough for a 6x6 board to be solved relatively quick (approximately 20 seconds), it was simply too slow for any larger tables. This became apparent when I let an 8x8 table run for almost ten hours before I gave up and canceled the operation. The reason behind this became apparent once I did the big O math on DFS. DFS is $O(b^d)$ where b is the branching factor of the graph and d is the depth of the goal state. b at the worse case is equal to 8, and d is equal the number of squares in the board, which in the case of an nxn table is n^2. So DFS of this application would be $O(8^{n^2})$. Yikes!

   However there is a Heuristic called Warnsdorf's rule which states that to find a open Knight's Tour in linear time, you simply pick the neighbor with the fewest neighbors for your next node. Applying this Heuristic to a DFS lowers the average branching factor b substantially by removing the randomness inherent in DFS and giving it a good Heuristic to get most of the work done.

2. There are three pieces of Python script in this tarball, dfs.py, wRule.py, and stack.py. dfs.py is a mock up of Depth First Search without Warnsdorf's Rule. It is just a referential piece of code that I wanted to keep in case I couldn't get the full version to work. wRule.py is the final code. stack.py contains a stack data structure I coded up over the summer so I would have it when I needed it.

3. There are 26,534,728,821,064 tours.

	A	B	C	D	E	F	G	H
1	35	32	39	50	23	62	19	52
2	38	25	36	33	40	51	22	63
3	31	34	49	24	61	20	53	16
4	26	37	30	41	54	43	64	21
5	29, 48, 27, 60, 1, 58, 17, 44							
6	4	7	10	47	42	55	14	57
7	9	28	5	2	59	12	45	16
8	6	3	8	11	46	15	56	13

Figure 1: One possible solution to the knight's tour on an 8x8 board.