Trace Of Iterative Deepening Search

1. userInput.py prompts to select search, choice is 1
2. userInput.py prompts to input string representing solitaire game (ex. < - - 0 0 0 - -, - - 0 X 0 - - , 0 0 X X X 0 0, 0 0 0 X 0 0 0, 0 0 0 X 0 0 0, - - 0 0 0 - -, - - 0 0 0 - - >)
3. Calls IterativeDeepeningSearch.iterativeDeepeningSearch(input)
4. Function sees if input is answer (functionsForBothSearches.isanswer(input)) or blank (functionsForBothSearches.checkIfXs(input))
5. Initializes list with input (as list of dictionary with attributes of “INPUT” and “MOVES”) (listOfNodes)
6. Goes into loop that continues until answer is either found or there is nothing else to explore
7. Produce children of first node (functionsForBothSearches.nextMoves(input))
8. Create the array of moves needed to get to each child
9. Make lists of dictionary for each child (input and moves)
10. See if any child is the solution
11. If it is, goes to functionsForBothSearches.printSolutions(solution, i, inputAnswer) and functionsForBothSearches.printAdditional(startTime, numberOfNodes, memory)
12. If not a solution, appends child list to listOfNodes
13. Repeats steps 7-12 until first list is completed (initial first list only contains one value, later they contain an entire set of children)
14. Removes first list in listOfNodes
15. Repeats steps 7-14 until listOfNodes is empty and there are no more nodes to explore
16. If empty, with go to functionsForBothSearches.printSolutions(solution, i, inputAnswer) and print such