CS 471 Final Project: Towers of Hanoi

By Joshua White, CS Major

**=============================================================**

**Check List (This must be checked off and included in your Cover Page):**

**1. \_Yes\_\_ Did you follow all the requirements in implementing the system?**

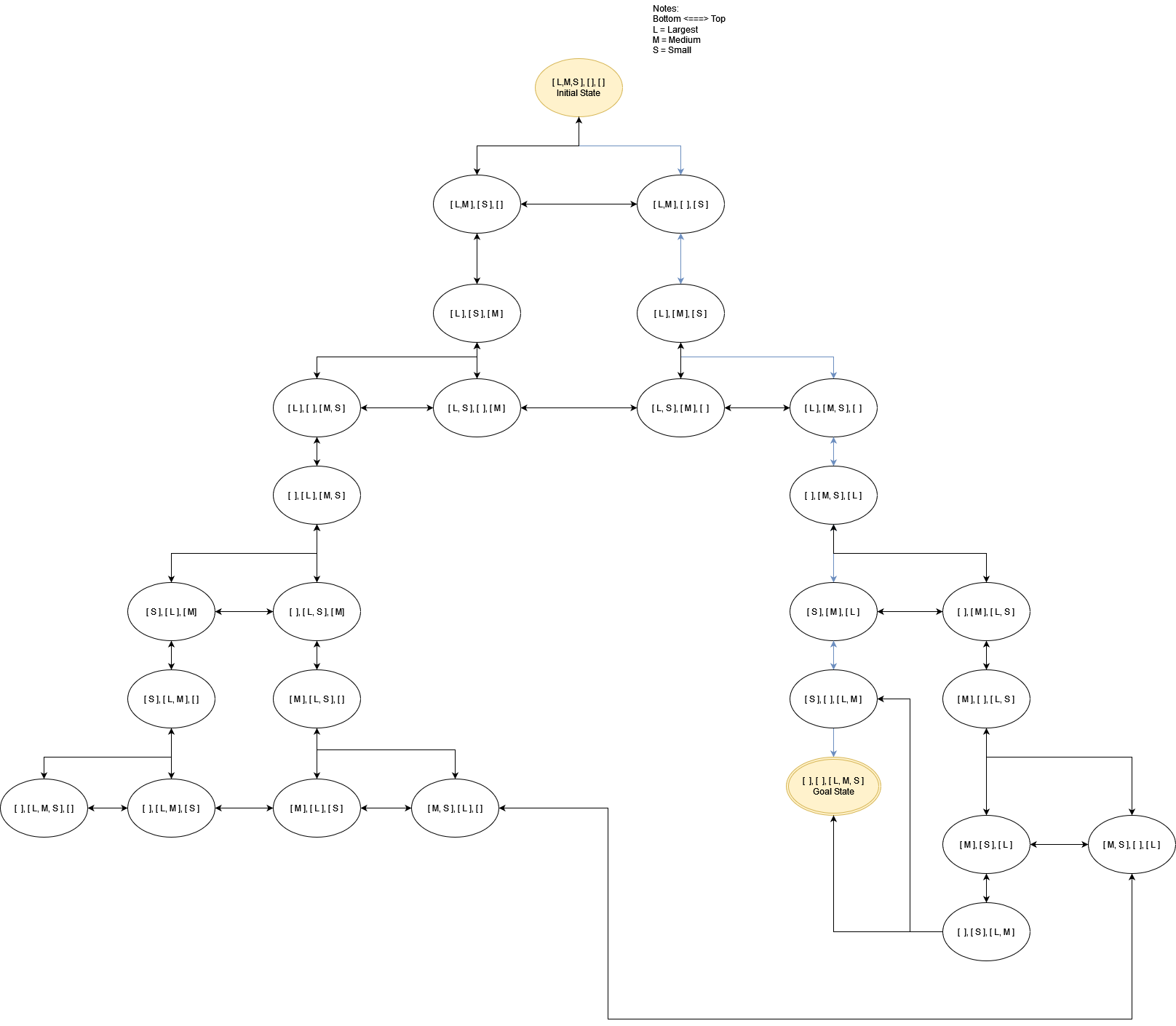
**2. \_\_Yes\_ Did you create one Word file of your report with the cover page and section headers as specified?**

**3. \_Yes\_\_ Did you answer all questions per section?**

==============================================================

**Section 1: Analysis of the Problem Space (6.1)**

1. List all possible disk move actions.
   1. Move Top Disk of Peg 1 to Peg 2
   2. Move Top Disk of Peg 1 to Peg 3
   3. Move Top Disk of Peg 2 to Peg 1
   4. Move Top Disk of Peg 2 to Peg 3
   5. Move Top Disk of Peg 3 to Peg 1
   6. Move Top Disk of Peg 3 to Peg 2
2. The Branching Factor is <= 6.
3. See graph
4. See graph
5. See graph



**Section 2: Designing the Evaluation Function for A\* (6.2)**

Assume that g = number of disk moves so far

h = estimate of how many more moves from a given state

f = g+h is the goodness

1. The equation for the estimate function h is: h = # of disks not in the correct spot.
2. Give an example h value and g value for 3 of the states in the above Problem Space.
   1. Initial State: [L,M,S], [ ], [ ], h = 3 and g = 0.
   2. State: [ ], [M,S], [L], h = 2 and g = 4
   3. State: [S], [ ], [L,M], h = 1 and g = 6.
3. Defend your decision for the h function:
   1. This is a good estimate of how many moves are required because it never overestimates. Example, State [L,M],[ ], [S] would have an h of 3 because the S is not in the correct position (on top of M which is on top of L).
   2. This h is admissible because it never overestimates.

**Section 3: Implementation (Source Code is submitted separately) (6.3)**

**Section 4: Testing and Results (6.4)**

1. Test your program **very** **thoroughly** and make sure the output matches your expectations.
2. Your analysis of the results:
   1. The Program worked as expected. It found a path to the goal.
   2. It took 12 disk moves to reach the goal.
   3. This is not optimal, since a person can do it in 7 disk moves.

**Section 5: Ideas for Adding Machine Learning**

1. Give one way to incorporate machine learning into the program to make it better?

* One way to incorporate machine learning is to include a means to learn what moves take away from the machine’s efficiency.
* This means to update the way it calculates f to include multiple factors.
* This learning would happen automatically at the end by observing the outcome of a game and comparing it to an efficient solution.

To be specific, this game which took 12 disk moves would look at the more efficient game which took 7.