Edwin Mak
CSE 373 HW#4

1) The worst case runtime of my union operation is O(1). By using weights, my union method simply adds the weights and sets a pointer by changing the number at the index.

2) My worst case runtime of my find operation is O(log N), but the amortized analysis of the find operation is very nearly O(1).

3) I tested the code using a test file that called the methods in MyDisjSets class. I also wrote a few up trees down on paper and coded them up to test their behavior.

4) n/a

5) My implementation could not produce a 1000 x 1000 maze in a reasonable time. I would consider a reasonable time to produce a 1000 x 1000 maze to be under 1 minute. I believe that there are a few parts of my program that delays the output of a larger maze. For example, initializing the internal edges takes O(N) time, as does randomly choosing indexes until the disjoint sets becomes one set. Printing the maze would also take O(N) time, and would require over 2 million prints. Using the ArrayLists from java collections might have also slowed down my code, since when I remove sets from virginMaze to put into inMazeSets the ArrayList must resize every time. If I were to improve this, I would simply take the item from the index I want and then swap values with the last item in the ArrayList.