

CSCI 262 Project 2 Write-Up

1. Explain your approach to solving this problem.

The first problem I had to solve was actually creating a list of tic tac toe boards. Thus, I decided to use a recursive function that repeatedly which in the end filled a vector element with every possible combination of x's, o's, and spaces in a nine character string. The recursive function loops through each position on the board, fills it with an x or o, and continues the recursion until it reaches a base case. The base cases of the function were when x wins, o wins, or there is a tie. Therefore, my program uses two functions, one that determines if a certain player won and another that determines if the board resulted in a tie, to test for these cases within the recursive function.

After my program was able to come up with all possible combinations of x's, o's, and spaces, I was able to both sort the data and eliminate any duplicate tic tac toe boards by passing the vector of strings into a set. I developed a function that counted the number of ties, the number of times x won, and the number of times o won. Also, I determined that the size of my set of strings was equal to the total number of tic tac toe boards possible. Finally, using the numbers I collected from my set of boards, I was able to print the required data to the screen.

2. Justify why the output of your program is *correct*.

The program is correct for a variety of reasons. In the beginning, I created the list of possible boards by determining every possible combination of x's and o's and spaces that could fill the board despite whether or not it was possible with the number of x's and o's each player is given. This means that I didn't miss any boards and I could accurately move on with the process used to eliminate any duplicate and invalid boards. Additionally, passing the data from the vector into a set ensured that I not only have a sorted list, but I also eliminated all duplicate boards from the set. In order to check the function that counts the total number of boards, ties, x wins, and o wins, I first printed the vector of board strings to a file and then compared my total number of boards to the number of lines in the file. Also before turning in my code, I checked my final numbers with the TA's in office hours and they matched which further improved my confidence that the output of my program is correct.

3. Consider the run time of your program; do you think your solution is efficient? Do you think it could be made more efficient?

My solution is efficient because it compiles and runs in a matter of a few seconds. However, I do think that it could have been made more efficient if I did not iterate through the long list of tic tac toe boards as many times in my program. If I could perform more of the necessary functions within one iteration, then I would increase the efficiency of the program and therefore, lower its run time.