MSSC 6000 - Midterm Exam

Henri Medeiros Dos Reis

March 21, 2023

For the problem number 1, I started by trying to find the whole search space for the problem, which I knew it was $(2n)!$, then I compared the length of my search space to $(2n)!$. After testing a bunch of different n values. After that, I tried to cut all the invalid results that were showing up in the search space by verifying if it violated the constraints. After making sure that no invalid answers were showing up in the search space, then it is as simple as just checking the score and grabbing the largest one. Then to test the actual results, I used some easy to check by hand data to test if the code was correct. Then, I tested against the provided input, and finally I tried to come up with some edge cases.

For the second problem, it was way easier to test, since it is a greedy algorithm. I just needed to check that all the answers that the algorithm would give would be valid. So I tried to code it in a way that all the constraints would not be violated. Then again, I tested using small n, so that I could find a solution by hand, then I tested against the provided input, and finally tried to come up with some edge cases.

For the third one I started by getting a solution from the brute force version, then I would run the backtracking and see if the solution that I was getting was correct. At the beginning I had a lot of bugs, so I used a debugger to see the what each variable was at each point of the code to check if what I thought that the code was doing was actually what the code was doing. Most of the bugs during tests were due to not searching all the valid answers, so once I fixed that, I went back to testing it versus the brute force version. Which gave correct results.