CS 370 - Final Project Selection

Group Members

- Aimal Wajihuddin
- Zach Saegesser
- Ryan Edelstein

Propsed Problems

1. **String Transmission**

On the surface, this seems like a very simple problems. But I think that first we would generate all of the different possible strings that could have been transmitted in the constraint of the allowed corrupted bits. After that, you would have to see if any of the ones we generated are periodic, and if they are, then get rid of them. The order of operations can/would change obviously, but that is the general approach I think. To be honest, I think that the biggest challenge with this would be to get it done in time.

1. **XOR Ninja**

The basic point of this problem is to generate the powerset of a set of numbers, and then XOR each set together. After gettng the results of all of the XORs, you would add them all together and return that. Similar to the last one, I believe the majority of the challenge with this would be getting it efficient enough to get it under the allotted time.

1. **What's Next**

I believe that this problem would require more of a strategic approach in terms of algorithm design. The point of this problem is to get the next number bigger than the one provided (in binary representation when put in order), that contains the same number of 1s as the one we got. The most basic approach I can immediately think of is to get that number and slowly add 1 to the number and count the number of 1s in that number until we get the desired output. That might be too inefficient and require optimizations though.

1. **A or B**

I think this is a **DP/use_it** or **lose_it** based algorithm. My initial guess is that you would start at the least significant bit (after converting the hexadecimal numbers to binary) and start changing bits in either A or B or both and go for a use_it or lose_it approach from there.