Tecnológico de Monterrey

ReflexActInt1_A01639373

Patricio Flores Reynoso A01645013

September 12, 2025

Analysis and Design of Advanced Algorithms

Gildardo Sanchez-Ante

To be honest, when I first saw the problem, I wasn't totally sure what algorithm would work best. But after some research and testing even with the constant feedback of my team, we ended up using the LPS array, which is part of the KMP algorithm. I didn't fully understand it at first, but once I saw how it avoids repeating checks when searching for patterns, it started to make sense. Basically, instead of checking every character again and again, it remembers where it left off and skips ahead. That makes it way faster, especially when the strings are long or have lots of repeated parts. I liked that it's predictable and doesn't rely on random stuff like hashing (which I read can mess up sometimes). We also aimed and tried to keep the code clean and organized by used vectors to store the transmissions and message codes, and made a function to read the files. That way, it's easier to add more test cases later or change things without breaking the whole program. While we know there are other algorithms like Rabin-Karp or Boyer-Moore, but they seemed more complicated or not as good for short patterns. So we stuck with LPS/KMP because it felt like the safest and most efficient option for what we needed.

Even though this was a team effort, I personally learned a lot from this activity. At first, I was just trying to follow the logic and get the code working, but as we progressed, I started to understand why certain decisions mattered. I realized that choosing the right algorithm isn't just about making things run—it's about making them run well, especially when the input gets bigger or more complex.

I also learned how important it is to break problems down. When I didn't understand the LPS array, I went step by step, trying to see what each part was doing. That helped me connect the dots and feel more confident about how the algorithm works. It's not just about memorizing code—it's about understanding the flow and why it's built that way.

This activity reminded me that it's okay not to get everything at once. What matters is staying curious, asking questions, and being open to learning from others. I still have a lot to learn, but now I feel more prepared to tackle similar problems and explore more advanced algorithms in the future.