Constance Xu, Hamzah Nizami, Eric Altenburg
I pledge my honor that I have abided by the Stevens Honor System.

What type of error you are getting on HackerRank. (You may "purchase" the input to the test case you are failing.)
What you tried to accomplish when coding the algorithm.
What test cases you ran locally and what is wrong with the output. If you see nothing wrong with any test cases locally, double check corner cases.
How you went about tracing/debugging your code.

In HackerRank, we were getting errors for different test cases. It varies every single time. But all of them pertain to timing out (the exact error that we get is "Your code did not execute within the time limits"). This was the only error that we continually got and it varied between which test cases we got it for (sometimes it would be for test case #6 or test case #11). We did not get any other errors aside from that one though. It also depended on whose computer we were using because on Hamzah's machine, HackerRank would have all of them pass, but on Eric and Connie's machine, that was not always the case. More often than not, it would work just fine.

When coding the algorithm, my group tried to get it to work as we are all fairly new to the C language. With this timing error, we tried to optimize where we could and we tried to implement it iteratively and we could not get it to work. This is why we decided to take a recursive approach which seemed more intuitive and it happened to work and we were able to pass the test cases in HackerRank; however, there were times when we did not pass the test cases and those were all due to timing out.

Test cases that we ran locally were the ones we were testing specifically for our search function. Our search function is the function that gave us the most trouble and we were trying to find intuitive ways to work with it. We added the words "Eric", "Erica", and "Erican" to see if we could search for the words "Ez", "eric", and "eri". We initially ran into some segmentation faults, but we were able to eliminate those problems by tracing on a whiteboard. Some corner cases that we checked were when we added "eric" and then added "q" to ensure that this would work. Short answer though, it did work.

Our group went about tracing our code by writing on a whiteboard and walking through each line when we were debugging specifically the search function (as this was the function that was giving us the most trouble). This was the most cumbersome part as most of us were just learning C for the first time (albeit a great challenge). Our bugs ranged from syntax errors to segmentation faults and we had a concrete idea of what we wanted to do before we started coding. It was difficult to figure out exactly what was going wrong at times, so we found ourselves just trying nearly everything to either come up with a working solution or finding a new way to do something. This took more time than we had expected to work on, which is understandable considering this is a new language for two members in the group. We drew out a lot of Tries on the whiteboard trying to figure out where our code was going wrong when it broke.