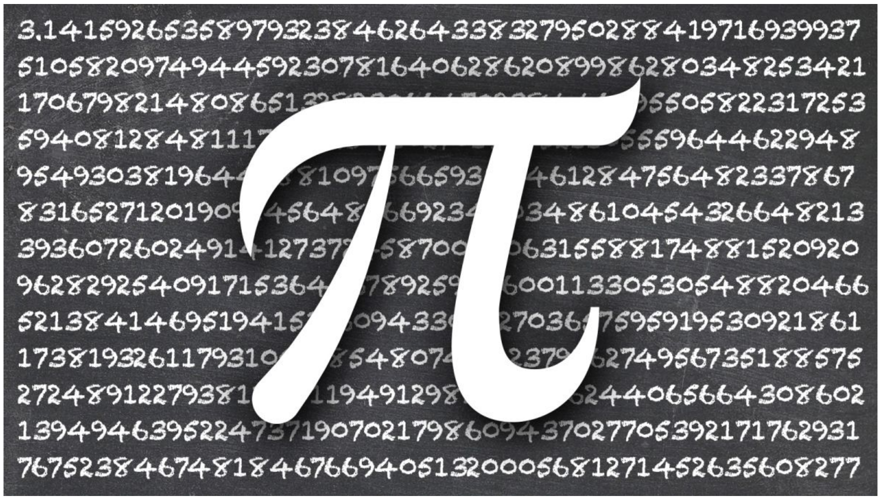
Introduction to Algorithms **F2021**

Problem Set 5

Handed Out: November 16^{th}, 2021 *Due: December 16^{th}, 2021*



In this (OPTIONAL) assignment, we will focus in on string matching. For the bonus portion of the previous assignment, I had you implement the Naive string matching algorithm. While this is a relatively simple algorithm, it's runtime leaves a lot to be desired. In this assignment, you'll implement the Rabin-Karp algorithm, a clever string matching algorithm that relies on hashing portions of the string in order to speed up the matching process.

For this assignment, we'll be working with the first 1000 digits of pi! I've included with this assignment one file: pi.txt. The first line of the file is a string containing the first 1000 digits of pi. The second line is a string of 5 characters that we'll use for matching.

Implement the Rabin-Karp Algorithm (40 Points)

Simply enough, all you have to do with this assignment is use the Rabin-Karp algorithm to determine where in the string of pi digits the search string occurs. The Rabin-Karp makes use of a hash function in order to make it easier to determine where the query string might occur in the larger string. For simplicity, please use the hash function we talked about in class (and that's used in the pseudocode). We'll use a d value of 10 and a q value of 61. In your writeup, please include how far into the string that the query pattern occurs (the shift value).

- For this assignment, you shouldn't need to use any high level data structures to implement the Rabin-Karp method. That being said, if you want to use a hash table, dictionary, or anything like that, you are free to do so.

- All code should be written in Python 3.8 OR C++. If you'd like to use another language, just ask and I will let you know if it's okay. There have been some unexpected errors that have occurred involving compiling code on the TA's machine. We've narrowed Python issues down to newer (Python 3.9+) distributions of Python. As such, please do not use Python 3.9+ to implement this assignment. For C++, be aware that we are compiling and running these assignments on Windows machines. Please ensure that your code runs on a Windows machine to ensure that things go smoothly. I'm looking into getting access to a Mac purely for grading purposes, but it's not clear that this will materialize in a timely manner.
- Please include instructions for how to compile and run your code in your writeup.
- Explain any implementation choices you had to make in the final report (such as underlying data structure), and also include where the pattern occurs in the search string (described above)!

Writeup (10 points)

You will include a written report with your submission detailing important details about your implementation, as well as the results of any analyses requested in the assignment. The report must be complete and clear. A few key points to remember:

- Complete: the report does not need to be long, but should include everything that was requested.
- Clear: your grammar should be correct, your graphics should be clearly labeled and easy to read.
- Concise: I sometimes print out reports to ease grading, don't make figures larger than they need to be. Graphics and text should be large enough to get the point across, but not much larger.
- Credit (partial): if you are not able to get something working, or unable to generate a particular figure, explain why in your report. If you don't explain, I can't give partial credit.