# Burrows–Wheeler

1. Please enroll in ADS - 2 Coursera by this link by signing in Coursera https://www.coursera.org/learn/algorithms-part2/
2. If your already enrolled in coursera of ADS - 2, go to this link which take you to problem statement https://www.coursera.org/learn/algorithms-part2/programming/3nmSB/burrows-wheeler/submission
3. Please follow the **instructions** before uploading zip files in Coursera.
    a. Implement the CircularSuffixArray. Be sure not to create explicit copies of the string (e.g., via the substring() method in Java's String data type) when you sort the suffixes. That would take quadratic space. Instead for each suffix, you only need to keep an index that indicates which character is the beginning of the suffix. This way you can build the *N* suffixes in linear time and space. Then sort this array of indices. It's just like sorting an array of references.
    b. Implement the Burrows-Wheeler transform, using the CircularSuffixArray class.
    c. The Burrows-Wheeler decoding is the trickiest part, but it is very little code once you understand how it works. (Not including declarations and input, our solution is about 10 lines of code.) You may find the key-indexed counting algorithm from the string sorting lecture to be useful.
    d. Implement the move-to-front encoding and decoding algorithms. Not including comments and declarations, our solutions are about 10 lines of code each. If yours is significantly longer, try to simplify it.

4. Your score must be greater than 80/100 to get 25/25 marks in Code camp assignment.
5. You need to manual push all the files below in github and submit commit id.
6. You need to submit the three source files **MoveToFront.java**, **BurrowsWheeler.java**, and **CircularSuffixArray.java** along with any other helper files needed to run your program.
7. Along with the above source file, please submit a **snapshot** of the assignment by the having a sample screenshot in this folder.