**Problem Statement:**

The Burrows–Wheeler data compression algorithm consists of three algorithmic components,

1.Burrows–Wheeler transform: The Burrows-Wheeler transform (BWT) is an algorithm that takes blocks of data, such as strings, and rearranges them into runs of similar characters. The nature of the algorithm tends to put similar characters next to each other, making the resulting data order easier to compress.

2. Move-to-front encoding: Given a text file in which sequences of the same character occur near each other many times, convert it into a text file in which certain characters appear much more frequently than others.

3. Huffman compression: Given a text file in which certain characters appear much more frequently than others, compress it by encoding frequently occurring characters with short codewords and infrequently occurring characters with long codewords.

**Related Concepts:**

Least Significant Digit (LSD) Radix Sort:

* A fast-stable sorting algorithm
* Begins at the least significant digit
* Proceeds to the most significant digit
* Lexicographic orderings

**Code:**

In this project, we have three files named as BurrowsWheeler.java, CircularSuffixArray.java, MoveToFront.java.

Burrows-Wheeler transform: Given a typical English text file, transform it into a text file in which sequences of the same character occur near each other many times.

Circular Suffix Array**: A suffix array is a sorted array of all suffixes of a given string.**

Move-to-front encoding and decoding: The main idea of move-to-front encoding is to maintain an ordered sequence of all of the characters, and repeatedly read in characters from the input message, print out the position in which that character appears, and move that character to the front.

**Test Cases:**

According to course era, compilation, API and spot bugs are passed, but pmd and check style passed with some warnings.

Correctness: 60/68 tests passed

Memory: 10/10 tests passed

Timing: 137/159 tests passed

Aggregate score: 90.17%

**Conclusion:**

I did the project and tested my code on course era.

Firstly, I got 0 and we did some modification in code. After trying, finally I got 90 score. But I couldn’t get to know about remaining errors showing in course era which were the reasons not to get 100 score.