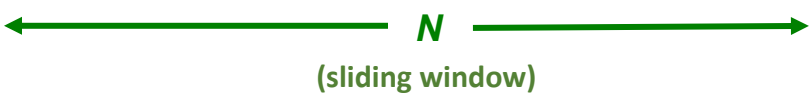


Project 1: Mining Data Streams

As we discussed the DGIM algorithms to estimate the number of 1's in the N most recent bits of the current position in a 1/0 stream. The current position (bit) is included as part of the N most recent bits (also known as a sliding window of size N) as illustrated below:

1001010110001011010101010101010110101010101110101010111010100010110010



N
(sliding window)

Consider the following paragraph of texts:

"In the 1990's "data mining" was an exciting and popular new concept. Around 2010, people instead started to speak of "big data." Today, the popular term is "data science." However, during all this time, the concept remained the same: use the most powerful hardware, the most powerful programming systems, and the most efficient algorithms to solve problems in science, commerce, healthcare, government, the humanities, and many other fields of human endeavor. To many, data mining is the process of creating a model from data, often by the process of machine learning, which we mention in Section 1.1.3 and discuss more fully in Chapter 12. However, more generally, the objective of data mining is an algorithm. For instance, we discuss locality-sensitive hashing in Chapter 3 and a number of stream-mining algorithms in Chapter 4, none of which involve a model. Yet in many important applications, the hard part is creating the model, and once the model is available, the algorithm to use the model is straightforward."

Generate a 1/0 stream from the above paragraph as follows:

1. Ignore all non-letter characters such as digits, dots, and spaces.
2. Convert each letter to its ASCII code (integer). You can refer to the following link for the ASCII table for conversion: <https://www.asciitable.com/>.
3. If the converted ASCII code is an odd number, the corresponding bit is 1; otherwise, the corresponding bit is 0.
4. The stream starts from the first letter and ends at the last letter.

You are asked to implement the DGIM algorithm with $N = 32$. Suppose your current bit is corresponding to the first letter ('s') of the last word ("straightforward") in the paragraph. Construct the buckets and estimate the number of 1's using the sliding window of given size ($N = 32$) for the current bit. Your DGIM algorithm should continue to estimate the number of 1's by taking new bits corresponding to the rest of the letters of the same word. Each new bit would generate a new count.

Submission Requirement:

1. There are 15 letters in the last word ("*straightforward*"), so your output should also contain 15 counts representing the number of 1's.
2. You may implement the DGIM algorithm in any programming language, but you should submit your source code together with a ReadMe file. In your ReadMe file, you need to explain how to compile and run your program. In addition, you should also include in the ReadMe the 15 counts generated by your program for the 15 letters of the last word ("*straightforward*"),.
3. All files should be submitted on Canvas.