**Assignment 1: Word Count with MapReduce Simulation**

**Topic:** MapReduce Programming Model (Lecture 9)  
**Time:** 30-40 minutes  
**Objective:** Simulate a simplified MapReduce process to count word frequencies in a text dataset.

**Problem Statement:**  
You are given a small text dataset (e.g., a paragraph from "Hamlet"). Write a Jupyter Notebook script to simulate the MapReduce process for counting word frequencies. Your script should include a map phase to split the text into words and emit (word, 1) pairs, and a reduce phase to sum the counts for each word.

**Dataset:**  
A simple text string provided in the notebook, e.g.,

python

text = "to be or not to be that is the question to be or not to be"

**Requirements:**

* Use Python lists and dictionaries (no external MapReduce framework).
* Implement a map\_function and a reduce\_function.
* Output a dictionary with words as keys and their frequencies as values.
* Visualize the top 5 most frequent words using matplotlib.

**Solution Outline:**

1. Define the map\_function that takes a string, splits it into words, and returns a list of (word, 1) tuples.
2. Define the reduce\_function that takes a list of (word, count) pairs, groups by word, and sums the counts.
3. Process the text: split into "shards" (e.g., sentences or chunks), apply map\_function, then reduce\_function.
4. Use pandas or collections.Counter to sort and extract the top 5 words.
5. Plot a bar chart with matplotlib.

**Sample Code Starter:**

python

CollapseWrapCopy

import matplotlib.pyplot as plt

from collections import Counter

text = "to be or not to be that is the question to be or not to be"

def map\_function(text\_chunk):

return [(word, 1) for word in text\_chunk.split()]

def reduce\_function(mapped\_data):

word\_counts = {}

for word, count in mapped\_data:

word\_counts[word] = word\_counts.get(word, 0) + count

return word\_counts

*# Students complete the rest*