**Question 2. Illustrate WordCount In-Mapper Combining Algorithm.**

Also assume that there are three input splits:

Input split 0 : [apple lemon mango salmon wheat apple mango]

Input split 1 : [barley salmon apple orange carrot rice salmon]

Input split 2 : [mango carrot lemon carrot apple rice tuna]

Since there are three input splits, there will be three Mappers. Thus, Input split I is handled by Mapper I (I = 0, 1, 2). Assume that there are three reducers. Note that Mapper I and Reducer I run on the same machine (I = 0, 1, 2).

Further, let the partitioner assign all words less than letter ‘k’ to Reducer 0, all words greater than ‘r’ to Reducer 2 and everything else to Reducer 1.

|  |  |  |  |
| --- | --- | --- | --- |
| [apple lemon mango salmon wheat apple mango] | [barley salmon apple orange carrot rice salmon] | [mango carrot lemon carrot apple rice tuna] | Input Split |
| (apple, 2), (lemon, 1), (mango, 2), (salmon, 1), (wheat, 1) | (barley, 1), (salmon, 2), (apple, 1), (orange, 1), (carrot, 1), (rice, 1) | (mango, 1), (carrot, 2), (lemon, 1), (apple, 1), (rice, 1), (tuna, 1) | Mapper Output |
| [apple, barley, carrot] | [lemon, mango, orange, rice] | [salmon, wheat, tuna] | S-S |
| (apple, [2, 1, 1])  (barley, [1])  (carrot, [1, 2]) | (lemon, [1, 1])  (mango, [2, 1])  (orange, [1])  (rice, [1, 1]) | (salmon, [1, 2])  (wheat, [1])  (tuna, [1]) | Reducer Input |
| (apple, 4)  (barley, 1)  (carrot, 3) | (lemon, 2)  (mango, 3)  (orange, 1)  (rice, 2) | (salmon, 3)  (wheat, 1)  (tuna, 1) | Reducer Output |