**Question 2. Illustrate WordCount Algorithm.**

Also assume that there are three input splits:

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

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

Input split 2 : [mango carrot lemon 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.

Answer:

|  |  |  |
| --- | --- | --- |
| {apple, lemon, mango, salmon, wheat, apple} | {barley, salmon, apple, orange, carrot, rice} | {mango, carrot, lemon, apple, rice, tuna} |
| (apple, 1) (lemon, 1) (mango, 1) (salmon, 1) (wheat, 1) (apple, 1) | (barley, 1) (salmon, 1) (apple, 1) (orange, 1) (carrot, 1) (rice, 1) | (mango, 1) (carrot, 1) (lemon, 1) (apple, 1) (rice, 1) (tuna, 1) |
| [apple, barley, carrot] | [lemon, mango, orange] | [rice, salmon, tuna , wheat] |
| S-S | | |
| Partition 0:  (apple, [1, 1, 1, 1])  (barley, 1)  (carrot, [1, 1]) | Partition 1:  (lemon, [1, 1])  (mango, [1, 1])  (orange, 1)  (rice, [1, 1]) | Partition 2:  (salmon, [1, 1])  (tuna, 1)  (wheat, 1) |
| (apple, 4)  (barley, 1)  (carrot, 2) | (lemon, 2)  (mango, 2)  (orange, 1)  (rice, 2) | (salmon, 2)  (tuna, 1)  (wheat, 1) |