BDT – cs523

Assignment 3 – MapReduce Basics

○ Submit your *own work* on time. No credit will be given if the assignment is submitted after the due date.
○ Note that the completed assignment should be submitted in .doc, .docx, .rtf or .pdf format only.

Assume that there are six input splits. Input splits 1,2 are on Machine 1, input splits 3,4 are on Machine 2 and input splits 5,6 are on Machine 3.

Input split1 : [cherry mango olive cherry]
[plum cherry banana cherry]

Input split2 : [cherry banana radish radish]
[pear banana mango cherry]

Input split3 : [banana kiwi plum banana]
[mango cherry kiwi banana]

Input split4 : [apple mango pear plum]
[radish kiwi banana olive]

Input split5 : [olive banana radish kiwi]
[cherry kiwi olive cherry]

Input split6 : [banana radish plum banana]
[olive cherry banana radish]

Also assume that there's only one reducer which is running on machine 1.

1. [9] **Illustrate the word count algorithm for the above scenario.**
Write your answer on the next page. (A table is already created for you)
Note: Illustrate means show mapper o/p, reducer i/p and reducer o/p.

   Remember to show the exact mapper output that gets stored locally.

2. [1] **How many tokens (key-value pairs) will be transferred across the network for getting the final reducer output?**
Ans: 48 tokens

Answer 1:

Machine 1		Machine 2		Machine 3	
Mapper 1 o/p for i/p split 1		**Mapper 3 o/p for i/p split 3**		**Mapper 5 o/p for i/p split 5**	
<cherry, 1> <mango, 1> <olive, 1> <cherry, 1>	<plum,1> <cherry, 1> <banana, 1> <cherry, 1>	<banana, 1> <kiwi, 1> <plum, 1> <banana, 1>	<mango, 1> <cherry, 1> <kiwi, 1> <banana, 1>	<olive, 1> <banana, 1> <radish, 1> <kiwi, 1>	<cherry, 1> <kiwi, 1> <olive, 1> <cherry, 1>
Mapper 1 – output file		**Mapper 3 – output file**		**Mapper 5 – output file**	
<banana, 1> <cherry, 1> <cherry, 1> <cherry, 1> <cherry, 1> <mango, 1> <olive, 1> <plum, 1>		<banana, 1> <banana, 1> <banana, 1> <cherry, 1> <kiwi, 1> <kiwi, 1> <mango, 1> <plum, 1>		<banana, 1> <cherry, 1> <cherry, 1> <kiwi, 1> <kiwi, 1> <olive, 1> <olive, 1> <radish, 1>	
Mapper 2 o/p for i/p split 2		**Mapper 4 o/p for i/p split 4**		**Mapper 6 o/p for i/p split 6**	
<cherry, 1> <banana, 1> <radish, 1> <radish, 1>	<pear, 1> <banana, 1> <mango, 1> <cherry, 1>	<apple, 1> <mango, 1> <pear, 1> <plum, 1>	<radish, 1> <kiwi, 1> <banana, 1> <olive, 1>	<banana, 1> <radish, 1> <plum, 1> <banana, 1>	<olive, 1> <cherry, 1> <banana, 1> <radish, 1>
Mapper 2 – output file		**Mapper 4 – output file**		**Mapper 6 – output file**	
<banana, 1> <banana, 1> <cherry, 1> <cherry, 1> <mango, 1> <pear, 1> <radish, 1> <radish, 1>		<apple, 1> <banana, 1> <kiwi, 1> <mango, 1> <olive, 1> <pear, 1> <plum, 1> <radish, 1>		<banana, 1> <banana, 1> <banana, 1> <cherry, 1> <olive, 1> <plum, 1> <radish, 1> <radish, 1>	
Shuffle & Sort					
Machine 1 Reducer input					

```
<apple, [1]>
<banana, [1,1,1,1,1,1,1,1,1,1,1]>
<cherry, [1,1,1,1,1,1,1,1,1,1]>
<kiwi, [1,1,1,1,1]>
<mango, [1,1,1,1]>
<olive, [1,1,1,1,1]>
<pear, [1,1]>
<plum, [1,1,1,1]>
<radish, [1,1,1,1,1,1]>
```

Reducer output:

Reducer output	
Apple	1
Banana	11
Cherry	10
Kiwi	5
Mango	4
Olive	5
Pear	2
Plum	4
Radish	6