Q1. Write an in-mapper combiner **algorithm** modifying algorithm 3.8 (That is, pairs approach)

class MapperPair

method initialize

H = new AssociativeArray

method map(doc id, doc d)

for all terms w in doc d do

for all terms u neighbors of w do

if(pair(w, u) == null)

H.put(pair(w, u), 1)

else

{H(pair(w, u)} = {H(pair(w, u)} + 1

method close

for all pair(w, u) in H do

Emit(pair(w, u), H{pair(w, u)})

**class Reducer**

method Reduce(pair p; counts [c1; c2; …])

s = 0

for all count c in counts [c1; c2; …] do

s = s + c .

**Emit**(pair p; count s)

**Q2. Write an in-mapper combiner algorithm modifying algorithm 3.9 (That is, stripes**

**approach)**

**class Mapper**

**method initialize**

M = new AssociativeArray

**method Map(docid a; doc d)**

for all term w in doc d do

H = new AssociativeArray

for all term u in Neighbors(w) do

H{u} = H{u} + 1

M{w} = sum(M{w}, H)

**method close**

for all term w in M

**Emit**(term w; Stripe M{w})

**class Reducer**

method Reduce(term w; stripes [H1;H2;H3; : : :])

Hf = new AssociativeArray

for all stripe H in stripes [H1;H2;H3; …] do

Sum(Hf; H) . //Element-wise sum

**Emit**(term w; stripe Hf )

Question 3

Data block 1 : [cat mat rat, cat]

Data block 2 : [cat bat cat pat]

Data block 3 : [cat bat rat bat]

Data block 4 : [cat rat bat rat]

Data block 5 : [bat mat pat bat]

Data block 6 : [pat cat bat mat]

a- Illustrate algorithm 3.8 (with no combiner, no in-mapper combining).

Data block 1, Data block 2 and Data block 3 are assigned to Mapper 1 -> Key, value pairs are like below:

|  |  |
| --- | --- |
| **Key,Value Pair** | **<K** |
| <(cat, mat), 1> | Not send |
| <(cat, rat), 1> | Not Send |
| <(mat, rat), 1> | Send to reducer2 |
| <(mat, cat), 1> | Send to reducer2 |
| <(rat, cat), 1> | Send to reducer2 |
| <(cat, bat), 1> | Not send |
| <(bat, cat), 1> | Not send |
| <(bat, pat), 1> | Not send |
| <(cat, pat), 1> | Not send |
| <(cat, bat), 1> | Not send |
| <(cat, rat), 1> | Not send |
| <(cat, bat), 1> | Not send |
| <(bat, rat), 1> | Not send |
| <(rat, bat>),1> | Send to reducer2 |

Data block 4, Data block 5 and Data block 6 are assigned to Mapper 2 🡪 (key, value) pairs are like below:

🡪

|  |  |
| --- | --- |
| **Key,Value Pair** | **<K** |
| <(cat, rat), 1> | Send to reducer1 |
| <(cat, bat), 1> | Send to reducer1 |
| <(cat, rat), 1> | Send to reducer1 |
| <(rat, bat), 1> | Not send |
| <(bat, rat), 1> | Send to reducer1 |
| <(bat, mat), 1> | Send to reducer1 |
| <(bat, pat), 1> | Send to reducer1 |
| <(mat, pat), 1> | Not send |
| <(mat, bat), 1> | Not send |
| <(pat, bat), 1> | Not send |
| <( pat, cat), 1> | Not send |
| <( pat, bat), 1> | Not send |
| <( pat, mat>),1> | Not send |
| <( cat, bat>),1> | Send to reducer1 |
| <( cat, mat>),1> | Send to reducer1 |
| <( bat, mat>),1> | Send to reducer1 |

|  |  |
| --- | --- |
| **Reducer1 Input** | **Reducer2 Input** |
| <(bat, cat), [1]>  <(bat, mat), [1,1]>  <(bat, pat), [1,1]>  <(bat, rat), [1,1]>  <(cat, bat), [1, 1, 1, 1, 1]>  <(cat, mat), [1,1]>  <(cat, pat), [1]>  <(cat, rat), [1 , 1 , 1, 1]> | <(mat, bat), [1]>  <(mat, cat), [1]>  <(mat, pat), [1]>  <(mat, rat), [1]>  <(pat, bat), [1,1]>  <(pat, cat), [1]>  <(pat, mat), [1]>  <(rat, bat), [1,1]>  <(rat, cat), [1]> |

🡪

|  |  |
| --- | --- |
| **Reducer1 Output** | **Reducer2 Output** |
| <(bat, cat), 1>  <(bat, mat), 2>  <(bat, pat), 2>  <(bat, rat), 2>  <(cat, bat), 5>  <(cat, mat), 2>  <(cat, pat), 1>  <(cat, rat), 4> | <(mat, bat), 1>  <(mat, cat), 1>  <(mat, pat), 1>  <(mat, rat), 1>  <(pat, bat), 2>  <(pat, cat), 1>  <(pat, mat), 1>  <(rat, bat), 2>  <(rat, cat), 1> |

b- Illustrate algorithm 3.8 (with in-mapper combining. That is, apply your algorithm Q1).

Data block 1, Data block 2 and Data block 3 are assigned to Mapper 1 -> Key, value pairs are like below:

|  |  |
| --- | --- |
| **Key,Value Pair** | **<K** |
| <(cat, mat), 1> | Not send |
| <(cat, rat), 2> | Not Send |
| <(mat, rat), 1> | Send to reducer2 |
| <(mat, cat), 1> | Send to reducer2 |
| <(rat, cat), 1> | Send to reducer2 |
| <(cat, bat), 3> | Not send |
| <(bat, cat), 1> | Not send |
| <(bat, pat), 1> | Not send |
| <(cat, pat), 1> | Not send |
| <(bat, rat), 1> | Not send |
| <(rat, bat>),1> | Send to reducer2 |

Data block 4, Data block 5 and Data block 6 are assigned to Mapper 2 🡪 (key, value) pairs are like below:

|  |  |
| --- | --- |
| **Key,Value Pair** | **<K** |
| <(cat, rat), 2> | Send to reducer1 |
| <(cat, bat), 2> | Send to reducer1 |
| <(rat, bat), 1> | Not send |
| <(bat, rat), 1> | Send to reducer1 |
| <(bat, mat), 2> | Send to reducer1 |
| <(bat, pat), 1> | Send to reducer1 |
| <(mat, pat), 1> | Not send |
| <(mat, bat), 1> | Not send |
| <(pat, bat), 2> | Not send |
| <( pat, cat), 1> | Not send |
| <( pat, mat>),1> | Not send |
| <( cat, mat>),1> | Send to reducer1 |

🡪

|  |  |
| --- | --- |
| **Reducer1 Input** | **Reducer2 Input** |
| <(bat, cat), [1]>  <(bat, mat), [2]>  <(bat, pat), [1,1]>  <(bat, rat), [1,1]>  <(cat, bat), [3, 2]>  <(cat, mat), [1,1]>  <(cat, pat), [1]>  <(cat, rat), [2, 2]> | <(mat, bat), [1]>  <(mat, cat), [1]>  <(mat, pat), [1]>  <(mat, rat), [1]>  <(pat, bat), [2]>  <(pat, cat), [1]>  <(pat, mat), [1]>  <(rat, bat), [1,1]>  <(rat, cat), [1]> |

🡪

|  |  |
| --- | --- |
| **Reducer1 output** | **Reducer2 output** |
| <(bat, cat), 1>  <(bat, mat), 2>  <(bat, pat), 2>  <(bat, rat), 2>  <(cat, bat), 5>  <(cat, mat), 2>  <(cat, pat), 1>  <(cat, rat), 4> | <(mat, bat), 1>  <(mat, cat), 1>  <(mat, pat), 1>  <(mat, rat), 1>  <(pat, bat), 2>  <(pat, cat), 1>  <(pat, mat), 1>  <(rat, bat), 2>  <(rat, cat), 1> |

c.

|  |  |
| --- | --- |
| Mapper 1 | Mapper 2 |
| Cat mat rat cat  Cat bat cat pat  Cat bat rat bat | Cat rat bat rat  Bat mat pat bat  Pat cat bat mat |
| (Cat,[( mat, 1),  (rat,1) ])  (mat, [(rat,1), (cat,1)])  (rat, [(cat,1)])  (cat, [(bat,1)])  (bat, [(cat,1),  (pat,1)])  (cat, [(pat,1)])  (cat,[( bat, 1), (rat,1), (bat, 1)])  (bat, [(rat, 1)])  (rat, [(bat, 1)]) | (cat, [(rat, 1), ( bat , 1),  (rat , 1)])  (rat, [(bat,1)])  (bat, [(rat,1)])  (bat, [(mat, 1), (pat,1)]  (mat , [(pat, 1), (bat,1)])  (pat, [(bat,1)])  (pat, [(cat, 1), (bat, 1), (mat,1)])  (cat, [(bat, 1), (mat, 1)])  (bat, [(mat, 1)]) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Partition

Words less "k" Otherwise

cat, bat mat, rat, pat

|  |  |
| --- | --- |
| **Reducer1 Input** | **Reducer2 Input** |
| (bat, [(cat,1),  (pat,1), (rat, 1),  (rat,1), (mat, 1), (pat,1), (mat, 1)])  (Cat,[( mat, 1),  (rat,1),(bat, 1), (pat, 1), (rat, 1) ( bat , 1), ( bat , 1), ( bat , 1), (rat, 1), ( bat , 1), (mat, 1), (rat, 1), ( bat , 1)]) | (mat, [(rat,1), (cat,1), (pat, 1), (bat,1)])    (pat, [(cat, 1), (bat, 1), (mat,1), ( bat , 1)])  (rat, [(cat,1), (bat, 1), (bat,1)]) |

|  |  |
| --- | --- |
| **Reducer1 output** | **Reducer2 output** |
| (bat, [(cat,1),  (pat,2), (rat, 2),  (mat, 2)])  (Cat,[( mat, 2),  (rat,4),(bat, 5), (pat, 1)]) | (mat, [(rat,1), (cat,1), (pat, 1), (bat,1)])    (pat, [(cat, 1), (bat, 2), (mat,1)])  (rat, [(cat,1), (bat, 2)]) |

D.

|  |  |
| --- | --- |
| Mapper 1 | Mapper 2 |
| Cat mat rat cat  Cat bat cat pat  Cat bat rat bat | Cat rat bat rat  Bat mat pat bat  Pat cat bat mat |
| (Cat,[( mat, 1),  (rat,2),(bat, 3), (pat, 1)])  (mat, [(rat,1), (cat,1)])  (rat, [(cat,1), (bat, 1)])  (bat, [(cat,1),  (pat,1), (rat, 1)]) | (cat, [(rat, 2), ( bat , 2), (mat, 1)])  (rat, [(bat,1)])  (bat, [(rat,1), (mat, 2), (pat,1)])  (mat , [(pat, 1), (bat,1)])  (pat, [(cat, 1), (bat, 2), (mat,1)]) |
|  |  |
|  |  |
|  |  |

partition

Words less than "k" Otherwise

cat, bat mat, rat, pat.

|  |  |
| --- | --- |
| **Reducer1 Input** | **Reducer2 Input** |
| (bat, [(cat,1),  (pat,1), (rat, 1),  (rat,1), (mat, 2), (pat,1)])  (Cat,[( mat, 1),  (rat,2),(bat, 3), (pat, 1), (rat, 2), ( bat , 2), (mat, 1)]) | (mat, [(rat,1), (cat,1), (pat, 1), (bat,1)])    (pat, [(cat, 1), (bat, 2), (mat,1)])  (rat, [(cat,1), (bat, 1), (bat,1)]) |

|  |  |
| --- | --- |
| **Reducer1 output** | **Reducer2 output** |
| (bat, [(cat,1),  (pat,2), (rat, 2),  (mat, 2)])  (Cat,[( mat, 2),  (rat,4),(bat, 5), (pat, 1)]) | (mat, [(rat,1), (cat,1), (pat, 1), (bat,1)])    (pat, [(cat, 1), (bat, 2), (mat,1)])  (rat, [(cat,1), (bat, 2)]) |