#### **Instructions**

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

Mapper

Method Initialize

H = new AssociativeArray

Method map (docID a, soc d)

for all w in d do

for all x in Neighbour (w) do

H{(w,u)}++

Method Close

for all (w,u) in H do

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

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

class Mapper

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

Emit(Term w; Stripe H)  
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) .

Emit(term w; stripe Hf )

Q3. Assume that there are two input spits and two reducers. Note that Mapper 1 and Reducer 1 run on the same machine. Mapper 2 and Reducer 2 run on the same machine.

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

Input Split 1 : [ {cat mat rat, cat}, {cat bat cat pat},{cat bat rat bat}] (Note : 3 records)

Input Split 2 : [{cat rat bat rat}, {bat mat pat bat}, {pat cat bat mat}] (Note: 3 records)

**Let the neighborhood of X, N(X) be set of all term after X and before the next X.**

Example: Let Data block be [a b c a d e]

N(a) = {b, c}, N(b) = {c, a, d, e}, N(c) = {a, d, e}, N(a) ={d, e}, N(d) = {e}, N(e) = {}.

1. Illustrate algorithm 3.8.
2. Illustrate algorithm 3.8 (with in-mapper combining. That is, apply your algorithm Q1.
3. Illustrate algorithm 3.9
4. Illustrate algorithm 3.9 (with in-mapper combining. That is, apply your algorithm Q2).