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

class Mapper

method Map(docid a; doc d)

for all term w in doc d do

for all term u in Neighbors(w) do

Emit(pair (w; u); count 1)

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 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 . //Tally words co-occurring with w

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) . //Element-wise sum

Emit(term w; stripe Hf )

**Q3. Assume that there are two mappers 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. Also assume that there are six data blocks:

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]

Data block 1, Data block 2 and Data block 3 are assigned to Mapper 1 and others to Mapper 2. 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 (with no combiner, no in-mapper combining).
2. Illustrate algorithm 3.8 (with in-mapper combining. That is, apply your algorithm Q1).
3. Illustrate algorithm 3.9 (with no combiner, no in-mapper combining).
4. Illustrate algorithm 3.9 (with in-mapper combining. That is, apply your algorithm Q2).