**Q.1 In-mapper Combiner algorithm – Pair Approach**

Class Mapper

Method Initialization

H = new Associative Array

Method map(docid a; doc d)

For all term w in doc d do

For all term u in neighbor(w) do

Pair p = new Pair(w, u)

H{p} = H {p} +1;

Method close

For all p in H do

Emit(p, H{p})

Class Reducer

Method reducer (pair p; counts [c1, c2, ..])

Sum = 0;

For all counts c in counts [c1, c2, …] do

Sum = Sum + c;

Emit(pair p; count c)

**Q.2 In-mapper combiner algorithm – Stripe approach**

Class Mapper

Method initialize

Hw = new Associative Array

Method map(docid a; doc d)

For all term w in doc d do

H = new Associative Array

For all term u in Neighbors(w) do

H{u} = H{u}+1

Hw{w} = sum(Hw{w}, H{u})

Method Close

For all tem w in Hw do

Emit(term w; Stripe H{w})

Class Reducer

Method reduce (term w; strips [H1, H2, H3..])

Hf = new Associative Array

For all stripes H in stripes [H1; H2; H3…]) do

Sum(Hf: H)

Emit(term w; stripe Hf)

**Q.3 (a):**

|  |  |  |
| --- | --- | --- |
| **0** | **1** | **Input-split (mapper input)** |
| [{cat mat rat cat}, {cat bat cat pat}, {cat bat rat bat}] | [{cat rat bat rat}, {bat mat pat bat}, {pat cat bat mat}] |
| **Record 1** | | **Mapper output** |
| ((cat, mat), 1)  ((cat, rat),1)  ((mat, rat), 1)  ((mat, cat), 1)  ((rat, cat), 1) | ((cat, rat),1)  ((cat, bat),1)  ((cat, rat), 1)  ((rat, bat), 1)  ((bat, rat), 1) |
| **Record 2** | |
| ((cat, bat), 1)  ((bat, cat), 1)  ((bat, pat), 1)  ((cat, pat),1) | ((bat, mat), 1)  ((bat, pat), 1)  ((mat, pat), 1)  ((mat, bat), 1)  ((pat, bat), 1) |
| **Record 3** | |
| ((cat, bat), 1)  ((cat, rat), 1)  ((cat, bat),1)  ((bat, rat),1)  ((rat, bat),1) | ((pat, cat), 1)  ((pat, bat), 1)  ((pat, mat), 1)  ((cat, bat), 1)  ((cat, mat), 1)  ((bat, mat), 1) |
| **Key <”k”** |  | **Shuffle & Sort** |
| ((bat, cat),[1])  ((bat, mat), [1,1])  ((bat, pat), [1, 1])  ((bat, rat),[1, 1])  ((cat, bat),[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]) |  |
| ((bat, cat), 1)  ((bat, mat), 2)  ((bat, rat), 2)  ((cat, bat), 2)  ((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) | **Reducer Output** |

**Q.3(b)**

|  |  |  |
| --- | --- | --- |
| **0** | **1** | **Input-split (mapper input)** |
| [{cat mat rat cat}, {cat bat cat pat}, {cat bat rat bat}] | [{cat rat bat rat}, {bat mat pat bat}, {pat cat bat mat}] |
| **Record 1** | | **Mapper output** |
| ((cat, mat), 1)  ((cat, rat),1)  ((mat, rat), 1)  ((mat, cat), 1)  ((rat, cat), 1) | ((cat, rat),2)  ((cat, bat),1)  ((rat, bat), 1)  ((bat, rat), 1) |
| **Record 2** | |
| ((cat, bat), 1)  ((bat, cat), 1)  ((bat, pat), 1)  ((cat, pat),1) | ((bat, mat), 1)  ((bat, pat), 1)  ((mat, pat), 1)  ((mat, bat), 1)  ((pat, bat), 1) |
| **Record 3** | |
| ((cat, bat), 2)  ((cat, rat), 1)  ((bat, rat),1)  ((rat, bat),1) | ((pat, cat), 1)  ((pat, bat), 1)  ((pat, mat), 1)  ((cat, bat), 1)  ((cat, mat), 1)  ((bat, mat), 1) |
| **Key <”k”** |  | **Shuffle & Sort** |
| ((bat, cat),[1])  ((bat, mat),[1,1])  ((bat, pat), [1, 1])  ((bat, rat),[1, 1])  ((cat, bat),[1,1])  ((cat, mat),[1,1])  ((cat, pat), [1])  ((cat, rat),[1,2,1]) | ((mat, bat), [1])  ((mat, cat),[1])  ((mat, pat), [1])  ((mat, rat),[1])  ((pat, bat), [1])  ((pat, cat),[1])  ((pat, mat),[1])  ((rat, bat), [1,1])  ((rat, cat), [1]) | **Reducer input** |
|  | | **Reducer output** |
| ((bat, cat), 1)  ((bat, mat), 2)  ((bat, rat), 2)  ((cat, bat), 2)  ((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) |

**Q.3 (c )**

|  |  |  |
| --- | --- | --- |
| **0** | **1** | **Input-split (mapper input)** |
| [{cat mat rat cat}, {cat bat cat pat}, {cat bat rat bat}] | [{cat rat bat rat}, {bat mat pat bat}, {pat cat bat mat}] |
| **Record 1** | | **Mapper output** |
| (cat, (mat:1, rat:1))  (mat, (rat:1, cat:1))  (rat, (cat:1)) | (cat, (rat:1, bat:1, rat:1))  (rat, (bat:1))  (bat, (rat:1)) |
| **Record 2** | |
| (cat, (bat:1))  (bat, (cat:1, pat:1))  (cat, (pat:1)) | (bat, (mat:1, pat:1))  (mat, (pat:1, bat:1))  (pat, (bat:1)) |
| **Record 3** | |
| (cat, (bat:1, rat:1, bat:1))  (bat, (rat:1))  (rat, (bat:1)) | (pat, (cat:1, bat:1, mat:1))  (cat, (bat:1, mat:1))  (bat, (mat:1)) |
| **Key <”k”** |  | **Shuffle & Sort** |
| (bat, ((rat:1), (cat:1, pat:1), (mat:1, pat:1), (rat:1), (mat:1))  (cat, (mat:1, rat:1), (rat:1, bat:1, rat:1), (bat:1), (pat:1), (bat:1, rat:1, bat:1), (bat:1, mat:1)) | (mat, (rat:1, cat:1), (pat:1, bat:1))  (pat, (bat:1), (cat:1, bat:1, mat:1))  (rat, (cat:1), (bat:1), , (bat:1)) | **Reducer Input** |
| (bat, (cat:1, mat: 2, pat:2, rat: 2))  (cat, (bat:2, mat: 3, pat: 1, rat:4)) | (mat, (bat: 1, cat: 1, pat:1, rat:1))  (pat, (bat:2, cat: 1, mat:1))  (rat, (bat:2, cat:1)) | **Reducer output** |

**Q.3 (d )**

|  |  |  |
| --- | --- | --- |
| **0** | **1** | **Input-split (mapper input)** |
| [{cat mat rat cat}, {cat bat cat pat}, {cat bat rat bat}] | [{cat rat bat rat}, {bat mat pat bat}, {pat cat bat mat}] |
| **Record 1** | | **Mapper output** |
| (cat, (mat:1, rat:1))  (mat, (rat:1, cat:1))  (rat, (cat:1)) | (cat, (bat:1, rat:2))  (rat, (bat:1))  (bat, (rat:1)) |
| **Record 2** | |
| (cat, (bat:1))  (bat, (cat:1, pat:1))  (cat, (pat:1)) | (bat, (mat:1, pat:1))  (mat, (pat:1, bat:1))  (pat, (bat:1)) |
| **Record 3** | |
| (cat, (bat:2, rat:1))  (bat, (rat:1))  (rat, (bat:1)) | (pat, (cat:1, bat:1, mat:1))  (cat, (bat:1, mat:1))  (bat, (mat:1)) |
| **Key <”k”** |  | **Shuffle & Sort** |
| (bat, ((rat:1), (cat:1, pat:1), (mat:1, pat:1), (rat:1), (mat:1))  (cat, (mat:1, rat:1), (bat:1, rat:2), (bat:1), (pat:1), , (bat:2, rat:1), (bat:1, mat:1)) | (mat, (rat:1, cat:1), (pat:1, bat:1))  (pat, (bat:1), (cat:1, bat:1, mat:1))  (rat, (cat:1), (bat:1) , (bat:1)) | **Reducer Input** |
| (bat, (cat:1, mat: 2, pat:2, rat: 2))  (cat, (bat:2, mat: 2, pat: 1, rat:4)) | (mat, (bat: 1, cat: 1, pat:1, rat:1))  (pat, (bat:2, cat: 1, mat:1))  (rat, (bat:2, cat:1)) | **Reducer output** |