# COMP9313 Assignment

z5126586 ZHENG FUDI

## Question 1:

**class Pair**

**firstNode, secondNode**

**int compareTo(Pair p)**

**int result = this.firstNode.compareTo(p.getFirstNode)**

**if (result == 0) result = this.secondNode.compareTo(p.getSecondNode)**

**return result**

**class NodesPair**

**firstNode, secondNode**

**int compareTo(Pair nodes\_p)**

**int result = this.firstNode.compareTo(nodes\_p.getFirstNode)**

**if (result ==0 ) result = this.secondNode.compareTo(nodes\_p.getSecondNode)**

**return result**

**class Mapper**

**method Map(tempNode, neighbor\_List)**

**tempArray = Array()**

**for close\_node in neignbour\_List**

**if tempNode < close\_node**

**tempPair = pair(tempNode , close\_node)**

**else**

**tempPair= pair(close\_node, tempNode)**

**tempArray.append(tempPair)**

**for first\_p in tempArray**

**for second\_p in tempArray**

**if first\_p != second\_p**

**emit(NodesPair(first\_p, second\_p),second\_p)**

**class Partitioner**

**method int getPartition(key, value, int numPartitions)**

**return key.first.hashCode() & Integer.MAX\_VALUE % numPartitions**

**Class NodesPairGroupingComparator extends WritableComparator**

**method int compare(WritableComparable term1, WritableComparable term2)**

**return ((NodesPair) term1).getFirst().compareTo(((NodesPair) term1).getFirst())**

**class Reducer**

**method Reduce(nodes\_pair, pairsLists)**

**emit(nodes\_pair.getFirst, concatenate(pairsLists))**

## Question 2:

**(1)**

**The 2-shingles for A and B is below:**

**Set A = {the sky, sky is, is blue, blue the, the sun, sun is, is bright};**

**Set B = {the sun, sun in, in the, the sky, sky is, is bright}.**

**We can get that, , ,**

**Therefore, .**

**(2)**

**We can get that , .**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **h2(n)** | **h1(n)** | **shingles** | **A** | **B** |
| **1** | **8** | **the sky** | **1** | **1** |
| **3** | **4** | **sky is** | **1** | **1** |
| **5** | **0** | **is blue** | **1** | **0** |
| **7** | **5** | **blue the** | **1** | **0** |
| **0** | **1** | **the sun** | **1** | **1** |
| **2** | **6** | **sun is** | **1** | **0** |
| **4** | **2** | **is bright** | **1** | **1** |
| **6** | **7** | **sun is** | **0** | **1** |
| **8** | **3** | **in the** | **0** | **1** |

|  |  |  |
| --- | --- | --- |
|  | **A** | **B** |
| **h1(n)** | **0** | **1** |
| **h2(n)** | **0** | **0** |