Q1

Pseudocode:

```
class LineGraphVertex
    // Pair of vertices in original graph
    method LineGraphVertex(i, j)
        set(i, j)

    // rewrite set method to ensure the 1st element i
    // is no larger than the 2nd element j
    // in line graph vertex (i, j)
    method set(i, j)
        this.i = min(i, j)
        this.j = max(i, j)

    // override compareTo method to enable secondary sort:
    method compareTo(LineGraphVertex lgv)
        cmp = this.i - lgv.i
        if cmp == 0
            cmp = this.j - lgv.j
        return cmp

class Mapper
    method map(src, destArray)
        // create all combinations of "line graph vertices":
        lgvList = new ArrayList()
        for each d in destArray
            lgvList.add(new LineGraphVertex(src, d))

        // emit new adjacency of "line graph vertices":
        for each lgv in lgvList
            Emit(lgv, lgvList except lgv)

class Reducer
    method reduce(lgv, list of lgvLists)
        lgvList = flatten list of lgvLists
        sortedLgvList = sortByv1(sortByv2(lgvList))
        Emit(lgv, sortedLgvList)
```

Q2

docA: "the sky is blue the sun is bright"
docB: "the sun in the sky is bright"

(i)

2-shinggles

```
docA: {"the sky", "sky is", "is blue", "blue the", "the sun", "sun is" "is bright"}
docB: {"the sun", "sun in", "in the", "the sky", "sky is", "is bright"}
```

	docA	docB
“the sky”	1	1
“sky is”	1	1
“is blue”	1	0
“blue the”	1	0
“the sun”	1	1
“sun is”	1	0
“is bright”	1	1
“sun in”	0	1
“in the”	0	1

Table 1. Bit vectors

Jaccard similarity

$$\text{jaccard}(docA, docB) = \frac{|docA \cap docB|}{|docA \cup docB|} = \frac{4}{9}$$

(ii)

Since the hash function is given:

$$h_1 = 5n - 1, \quad h_2 = 2n + 1$$

The input matrix with bit vectors and corresponding hash value is:

	docA	docB	h1	h2
Row 0	1	1	8	1
Row 1	1	1	4	3
Row 2	1	0	0	5
Row 3	1	0	5	7
Row 4	1	1	1	0
Row 5	1	0	6	2
Row 6	1	1	2	4
Row 7	0	1	7	6
Row 8	0	1	3	8

Table 2. Input matrix with corresponding hash values

Signature matrix can be derived from following steps:

		docA	docB
Initialize	h1	∞	∞
	h2	∞	∞
Row 0	h1	8	8
	h2	1	1
Row 1	h1	4	4
	h2	1	1
Row 2	h1	0	4
	h2	1	1
Row 3	h1	0	4
	h2	1	1
Row 4	h1	0	1
	h2	0	0
Row 5	h1	0	1
	h2	0	0
Row 6	h1	0	1
	h2	0	0
Row 7	h1	0	1
	h2	0	0
Row 8	h1	0	1
	h2	0	0

Table 3. Steps for One Pass implementation of Min-hashing

Hence, the signature matrix is

DocA	DocB
0	1
0	0

Table 4. Signature matrix