Main Idea

We run a inner join on node using two relations. One contains out neighbors and another contains in neighbors. Cross join these two will leads to 2 hop in neighbors connected through current key node.

Mapper

```
Foreach src, target pairs:
    emit(src, tgt, 1)
    emit(tgt, src, 2)
```

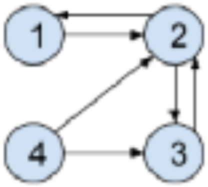
Reducer

```
out = filter input with 3rd element equals 1
in = filter input with 3rd element equals 2

for _ src, _ in in:
    for _, tgt, _ in out:
        if src != tgt:
            emit (tgt, src)
```

Trace

To make it clearer, we have below trace for the toy input graph



Mapper

Input:

src	tgt
4	3
1	2
2	3
4	2
2	1
3	2

output:

4	3	1
3	4	2
1	2	1
2	1	2
2	3	1
3	2	2
4	2	1
2	4	2
2	1	1
1	2	2
3	2	1
2	3	2

Reducer

After shuffle of Mapper output, reducer has below input. Tuples to same reducer was put in same row.

```
1: 1   2   1, 1   2   2
2: 2   3   1, 2   4   2, 2   1   1, 2   3   2
3: 3   4   2, 3   1   2, 3   2   2, 3   2   1
4: 4   2   1, 4   3   1
```

The in and out for each key is

```
1: out = (1   2   1), in = (1   2   2)
2: out = (2   3   1, 2   1   1), in = (2   4   2, 2   3   2, 2 1 2)
3: out = (3   2   1), in = (3   4   2, 3   2   2)
4: out = (4   2   1, 4   3   1)
```

For each key we have output

```
1:
2: 3 4, 3 1, 1 3, 1 4
3: 2 4
4:
```

Put it together, the final output is

```
3 4
3 1
1 3
1 4
2 4
```