Solution Q3d: Illustrate algorithm 3.9 (with in-mapper combining. Apply your algorithm Q2).

<table>
<tr><td>INPUT</td><td>Input Split-1</td><td>Input Split-2</td></tr>
<tr><td>Mapper Input</td><td>
<table>
<tr><td>cat mat rat cat</td></tr>
<tr><td>cat bat cat pat</td></tr>
<tr><td>cat bat rat bat</td></tr>
</table>
Neighbours:

N(cat) = {mat,rat}
N(mat) ={rat,cat}
N(rat) = {cat}
N(cat) = {}

N(cat) = {bat}
N(bat) = {cat,pat}
N(cat) = {pat}
N(pat) = {}

N(cat) = {bat,rat,bat}
N(bat) = {rat}
N(rat) = {bat}
N(bat) = {}
</td><td>
<table>
<tr><td>cat rat bat rat</td></tr>
<tr><td>bat mat pat bat</td></tr>
<tr><td>pat cat bat mat</td></tr>
</table>
Neighbours:

N(cat) = {rat,bat,rat}
N(rat) = {bat}
N(bat) = {rat}
N(rat) = {}

N(bat) = {mat,pat}
N(mat) = {pat,bat}
N(pat) = {bat}
N(bat) = {}

N(pat) = {cat,bat,mat}
N(cat) = {bat,mat}
N(bat) = {mat}
N(mat) = {}
</td></tr>
<tr><td>MAP</td><td>Mapper-1</td><td>Mapper-2</td></tr>
<tr><td>Mapper Output</td><td>(cat, { mat:1, rat:2, bat:3, pat:1 })

(mat, { rat:1, cat:1 })

(rat, { cat: 1, bat:1 })

(bat, { cat:1, pat:1, rat:1 })</td><td>(cat, { rat:2, bat:2, mat:1 })

(rat, { bat:1 })

(bat, { rat:1, mat:2, pat:1 })

(mat, { pat:1, bat:1 })

(pat, { bat:2, cat:1, mat:1 })</td></tr>
</table>

PARTITION	**(a-j)**	**(k-z)**
	(cat, { mat:1, rat:2, bat:3, pat:1 }) (bat, { cat:1, pat:1, rat:1 }) (cat, { rat:2, bat:2, mat:1 }) (bat, { rat:1, mat:2, pat:1 })	(rat, { bat:1 }) (mat, { pat:1, bat:1 }) (pat, { bat:2, cat:1, mat:1 }) (mat, { rat:1, cat:1 }) (rat, { cat: 1, bat:1 })
SORT & COMBINE		
Reducer Input	(bat, [{ cat:1, pat:1, rat:1 }, { rat:1, mat:2, pat:1 }]) (cat, [{ rat:2, bat:2, mat:1 }, { mat:1, rat:2, bat:3, pat:1 }])	(mat, [{ pat:1, bat:1 }, { rat:1, cat:1 }]) (pat, [{ bat:2, cat:1, mat:1 }]) (rat, [{ cat: 1, bat:1 }, { bat:1 }])
REDUCE	**Reducer-1**	**Reducer-2**
Reducer Output	(bat, { cat:1, mat:2, pat:2, rat:2 }) (cat, { bat:5, 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 })