

Solution Q3c: Illustrate algorithm 3.9.

INPUT	Input Split-1	Input Split-2
Mapper Input	<div><div><div>cat mat rat cat</div><div>cat bat cat pat</div><div>cat bat rat bat</div></div><div><div>Neighbours:</div><div>N(cat) = {mat, rat}</div><div>N(mat) = {rat, cat}</div><div>N(rat) = {cat}</div><div>N(cat) = {}</div><div>N(cat) = {bat}</div><div>N(bat) = {cat, pat}</div><div>N(cat) = {pat}</div><div>N(pat) = {}</div><div>N(cat) = {bat, rat, bat}</div><div>N(bat) = {rat}</div><div>N(rat) = {bat}</div><div>N(bat) = {}</div></div></div>	<div><div><div>cat rat bat rat</div><div>bat mat pat bat</div><div>pat cat bat mat</div></div><div><div>Neighbours:</div><div>N(cat) = {rat, bat, rat}</div><div>N(rat) = {bat}</div><div>N(bat) = {rat}</div><div>N(rat) = {}</div><div>N(bat) = {mat, pat}</div><div>N(mat) = {pat, bat}</div><div>N(pat) = {bat}</div><div>N(bat) = {}</div><div>N(pat) = {cat, bat, mat}</div><div>N(cat) = {bat, mat}</div><div>N(bat) = {mat}</div><div>N(mat) = {}</div></div></div>
MAP	Mapper-1	Mapper-2
Mapper Output	<div><div>(cat, { mat:1, rat:1 })</div><div>(mat, { rat:1, cat:1 })</div><div>(rat, { cat: 1 })</div><div>(cat, { bat:1 })</div><div>(bat, { cat:1, pat:1 })</div><div>(cat, { pat: 1})</div><div>(cat, { bat:1, rat:1, bat:1 })</div><div>(bat, { rat:1 })</div><div>(rat, { bat:1 })</div></div>	<div><div>(cat, { rat:1, bat:1, rat:1 })</div><div>(rat, { bat:1 })</div><div>(bat, { rat:1 })</div><div>(bat, { mat:1, pat:1 })</div><div>(mat, { pat:1, bat:1 })</div><div>(pat, { bat:1 })</div><div>(pat, { cat:1, bat:1, mat:1 })</div><div>(cat, { bat:1, mat:1 })</div><div>(bat, { mat:1 })</div></div>

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