```
map1(x,y): write(x,y)
reduce1(x, L):
  for each y in L do write(y, (x,L/{y}))
```

```
map2(x,y): write(x,y)
reduce2(x, L):
  NeighborsOfX (list) = {}
  NeighborsOfNeighbors (map) = {}
  edges = 0
  for each (y,z1 z2 z3 ...) in L do
    NeighborsOfX.add(y)
    NeighborsOfY (list) = {}
    NeighborsOfY.add(z1, z2, z3, ...)
    NeighborsOfNeighbors.put(y, neighborsOfY)
  for each y in NeighborsOfX do
    for each n in NeighborsOfY do
      if n in NeighborsOfX then edges = edges + 1
  edges = edges / 2
  d = |NeighborsOfX|
  write(x, d>1 ? 2edges/(d(d-1)) : 0)
```