Map_1 Input : (Author, Paper, Citations)
Output : (Paper, Citations, Author)

key — {works if citations not key}

$Reduce_1$ Input : (Paper, Citations, $\{A_1, \ldots, A_n\}$) — a bag, not a set; A_i = Author
Output : $(A_i, A_j,$ Citations) where $i < j$

{Avoid duplicating pairs or having pairs (A_i, A_i)}

Map_2 Input : $(A_i, A_j,$ Citations)
Output : $(A_i, A_j,$ Citations)

$Reduce_2$ Input : $(A_i, A_j, \{C_1, \ldots, C_m\})$
Output : $(A_i, A_j, \sum_{i=1}^{m} C_i)$

Map_3 Input : (A_i, A_j, Σ)
Output : (Σ, A_i, A_j)

[Sort: descending comparator]

$Reduce_3$ Input : $(\Sigma, \{(A_{i1}, A_{j1}), \ldots (A_{in}, A_{jn})\})$ [just buffer to output]
Output : (Σ, A_{ik}, A_{jk}) $1 \le k \le n$

Map_1 / $Reduce_1$: Output co-authors and citations per paper

Map_2 / $Reduce_2$: Sum citations for each pair of coauthors across all papers

Map_3 / $Reduce_3$: Sort the co-authors/citations in desc. order by citations

Combiner? Only for $Reduce_2$

Map_{1A} Input: (Receipt, Item)
Output: (Receipt, Item)

Map_{1B} Input: (Receipt, Time)
Output: (Receipt, Time)

$Reduce_1$ Input: (Receipt, $\{I_1, \ldots I_n, T\}$)
Output: (I_j, T) $1 \leq j \leq n$

Map_{2A} Input: (I, T)
Output: (I, T)

Map_{2B} Input: (Item, Name, Price)
Output: (I, P)

$Reduce_2$ Input: $(I, \{T_1, \ldots, T_n, P\})$
Output: $(H_\#, P'_\#)$
where $H_\#$ is an hour,
$P'_\#$ is the total value of item I sold in H
$P'_\# = (C \times P)$ where C is number of T_i in H

Map_3 Input: $(H, P'_\#)$
Output: (H, P') ← can use a combiner

$Reduce_3$ Input: $(H, \{P'_1, \ldots, P'_n\})$
Output: $(H, \sum P'_i)$ $1 \leq i \leq n$

Map_4 / $Reduce_4$: sort descending by $\sum P'$

Map_{1A} / Map_{1B} / $Reduce_1$: Get time each item was sold
Map_{2A} / Map_{2B} / $Reduce_2$: Get price of each item and sum for each hour
Map_3 / $Reduce_3$: Sum all item values for each hour
Map_4 / $Reduce_4$: Sort

Alternative method: Join tables 1 and 3 first to get price of each item on each receipt, sum prices for each receipt, sum receipts for each hour