

1. Suppose your input is a collection of key-value pairs where the key is a pairId and the value is (u, f), indicating that f follows u (as in Twitter). Describe separate map-reduce programs that produce each of the following desired outputs:
 - (a) A collection of key-value pairs (u, C) where u is a user and C is the number of followers.
 - (b) A collection of pairs (u, fof) such that u is a user and fof is a follower of someone who follows u. *Hint: suppose (a, b) and (b, c) are in the original input, your program needs to bring a and c together to construct (a, c) in the result. Keep in mind that (c, a) should not necessarily appear in the result – you need to not only bring them together but distinguish the person from the follower of the follower.*
 - (c) A collection of key-value pairs ((u1, u2), C) where the key is (u1, u2) and the value is C, the number of people who follow *both* u1 and u2. *Hint: you may need to write multiple map-reduce programs.*
2. Suppose we execute word count program on large collection of documents (say 100,000) with M map tasks and R reduce tasks. (Recall how data is sent from map tasks to reduce tasks.) Let's fix $M = 10,000$. As described in class, the map program emits (word, 1) pairs for each word and the reduce program takes a list of ones for each word and emits (word, SUM) where SUM is the sum of the list of ones.
 - (a) Suppose no combiner is used and $R = 10,000$. Do you expect significant skew in the times taken by the various reducers to process their value list?
 - (b) Suppose no combiner is used and $R = 10$. Do you expect significant skew?
 - (c) Suppose a combiner is used and $R = 10,000$. Do you expect significant skew?