

Problem a)

- 1) For Map process, input is the line of data, such as 117 51 1. This process will read line by line and split it into array to store data. To obtain the first term and second term as key of reduce process and its weight for each pair.
- 2) For Map reduce process, it will read all Map <key, value> pair from map process and find the maximum value of each key. For each key, we will obtain its max value and output it as final answer.

Overall Process: Original Data, line <src,tgt,weight> (n lines), then Map <key,weight> (2n pair) and final Reduced <key,max_weight>

Trace:

- 1) Input file
- 2) Output of Map
(117,1); (51,1); (194,1); (51,1); (299,3);(51;3) and so on
- 3) Map Reduce will read above data and group them with same Key value such as (51,1);(51;3) and (51,1) and Find biggest value as final output
- 4) Output: Like final output

Problem b)

Algorithm:

- 1) To simplify computation, first step is grouping initial input with ascending order of source and target order.

Out put like:

(1 2)(2 1)(2 3)(3 2)(4 2)(4 3) (Actually, not sort is also ok for this algorithm but needs more step for finding)

- 2) Then for each line, read its data in map step and store this pair
- 3) For each pair of data such as (1 2), obtain its second value 2, find all pairs with first number 2. (2 1)(2 3)
- 4) Map process will lease sorted graph.
- 5) If Second found pair's last value equals to 1, discard it. Else output <Second(2), First(1)> as final result. (Reduce process)
- 6) (1 2), (2 1) will be discard and (2 3) remains. Final answer is (3 1)
(2 1), only (1 2) available but failed. (2 3), only(3 2) but failed. Like this process
- 7) From top to bottom, group all results will be the final answer