Assignment 3

1.Describe your map/reduce algorithm for solving the three's company problem.

1. Describe the operation of the mapper and reducer. How does this combination solve the three's company problem?

   Ans.

   In our assignment, for a given set of numbers the first number is assigned as main friend (mf). Then, the loop is run for remaining numbers in selecting one subsequent value as key. For each key value a new set is formed which has the main friend and one of the remaining numbers (thus a set of three numbers) The other two numbers except the key value are arranged in ascending order.

   ```
   eg. Input: 4 2 3 5
       Mapper Output: 2 3 4
                      2 4 5
                      3 2 4
                      3 4 5
                      5 2 4
                      5 2 3
   ```

   In reducer for each key value is searched along with its values. For the triplet which is repeated the reducer prints the set. 2 1 3 4

   ```
   Eg. Input: 2 1 3 4
              3 2 1
       Mapper O/p: 1 2 3
                   1 2 4
                   3 1 2
                   3 2 4
                   4 1 2
                   4 2 3
                   2 1 3
                   1 2 3
       Reducer O/p  1 2 3
   ```

   The above example also shows solving of Three's Company problem.

2. What is the potential parallelism? How many mappers does your implementation allow for? Reducers?

   The case of running each input file on separate mapper simultaneously and all the entries with same key running on reducers simultaneously is known as potential parallelism.

   My implementation allows the maximum number of mappers or reducers to be the maximum number of friends.

3. What types are used as the input/output to the mapper? Motivate the transformation.

   Ans

   input:

   Object: Mapper Class reference.

   Text: Input data which has friend list for each file.

   Output:

   Text: Key of the mapper. It is text since it is a friend.

   Text: Value of the mapper. Being a pair of friends it is of datatype text.

2. On combiners
    1. Why did you leave the combiner class undefined in Step 4?
       Ans:
       The combiner class is there to combine repeated outputs of the mapper and thus pass less amount of data further. However, we use the reducer which only selects the outputs of the mapper which are repeated. If we combine the mapper output then the reducer would not be able to detect and thus print the repeated triplets.
    2. Generalize the concept: What sort of computations cannot be conducted in the combiner?
       Ans:
       Computations that are not commutative and associative cannot be conducted in the combiner. Computations of involving comparison, subtraction, division etc fall under this category.
3. Analyze the parallel and serial complexity of the problem and your M/R implementation (in Big-O notation). You should assume that there are n friends list each of length l, i.e. n users that each have l friends.
    1. What is the fundamental serial complexity of the problem? Think of the best serial implementation.
       The fundamental serial complexity of the problem is $O(nl^2)$. Even in best serial implementation, assuming n friends, we have to traverse through the loop of l length for each friend.
    2. How much work in total (over all mappers and reducers) does the Map/Reduce algorithm perform?
       As, the mappers and reducers do $O(nl^2)$ work separately, the total work can be given as $O(2nl^2)$.
    3. How much work is performed by each mapper? By each reducer?
       Each mapper or reducer traverses the length of the loop once and hence the work done is $O(l^2)$
    4. Based on your answers to the above, describe the tradeoff between complexity and parallelism (qualitatively, you have already quantified it in the previous steps).

       Complexity if the code increases with parallelism. The work becomes $O(2nl^2)$ in case of $O(nl^2)$ during serial implementation.