Homework Assignment 1

Problem 1

Question 1

Map phase: Saperate all Documents into several exclusive sets. Each set contains diffrent but complete sentences. Assign every set into a Map component, inside which are turples (word, count_of_sentences) calculated.

Shuffle phase: Group the turples with key of same word together and devide it into reducer according to words.

Reduce phase: For each reducer sum up number of sentences, in which each reducer assigned words occur.

at last combine results from all of reducers, output

Question 2

Map phase: Two mapper receive two lists of turple, in each of mapper is the list sorted according to the lexical order of words.

Shuffle phase: At first set a volume *n* of a reducer. Each reducer is assigned at most n (pairs of, if word duplicated in two lists) turple.

Reduce phase: Sum up count of each word.

Question 3

Map phase same as 2.. Just in Shuffle phase if a word only appears in one list the delete it. Reduce phase same as 2..

Problem 2

```
collection = LOAD 'wikipedia_mapreduce.txt' AS (text: chararray);

collection_with_ids = RANK collection;

/**
 *Here could it be implemented using a mapreduce, map the collection with
 *ids into several mappers and tokenize all words, shuffle and reduce do
 *basically nothing.
 **/
tokenized = FOREACH collection_with_ids GENERATE $0 AS doc_id, TOKENIZE(text) AS tokens;
```

```
/**
 *map the tokenized into several mappers and flatteen them, shuffle and
 *reduce do basically nothing.
 **/
tokens = FOREACH tokenized GENERATE doc_id, FLATTEN(tokens) AS token;

/**
 *map the tokens into several mappers and change them into lower cases,
 *shuffle and reduce phase could be done in following statement.
tokens_lower = FOREACH tokens GENERATE doc_id, LOWER(token) AS token;

/**
 *Shuffler group same token together and reducer remove duplicated
 *tokens.
 **/
tokens_distinct = DISTINCT tokens_lower;

/**
 *every FOREACH could be mapped into several mappers
 **/
tokens_grouped = FOREACH (GROUP tokens_distinct BY token) {
    /**
     *here is a 'sub' map procedure which inside each mapper could all
     *documents in which this token appears be destributed into several
     *'sub-mappers' and do the statistical stuff.
    doc_ids = FOREACH tokens_distinct GENERATE doc_id;
    doc_ids_sorted = ORDER doc_ids BY doc_id;
    GENERATE group AS token, doc_ids_sorted;
}

tokens_sorted = ORDER tokens_grouped BY token;
STORE tokens_sorted INTO 'results/pig/wikipedia_mapreduce_index';
```

Problem 3

Question 1

All of the numbers could be seperated into mappers, inside each mapper the largest number of current subset is found. Then use one reducer to find the largest number amoung all of numbers using result from mappers.

Question 2

Every mappers recieve a exclusive subset of whole number set and number of the elements contained in each subset and caculate the sum of each subset. A reducer add up all the sums from mappers as well as numbers of all element, then calculate the average.

Question 3

Inside each mapper all duplicated numbers are elimenated. Rest are numbers that are distinct inside each sub-set. They are sent to a reducer to reduce all the numbers which are inter-subset-duplicated.

Question 4

Map phase same as Question 3. In shuffle phase are all elements sorted and groupped by element value and sent to a reducer. In reducer are all groups which contain more than one element deleted.

Problem 4

Consider an MxN matrix A multiplicates with an NxP matrix B and result is an MxP matrix C, where by $c_{i,j}$=sum$_{\text{k for k in (1..N)}}$($a_{i,k}$*$b_{k,j}$)

We could assign every single mutiplication, i.e. $a_{i,k}$*$b_{k,j}$, into a mapper $M_{i,k,j}$, send the $a_{i,k}$ and $b_{k,j}$ into $M_{i,k,j}$.

Then in reduce phase every result from $M_{i,k,j}$ with same i and j are grouped together and sent to a reducer

Inside every reducer is each element of C, i.e. $c_{i,j}$ caculated by adding up all $a_{i,k}$ and $b_{k,j}$ (for k in (1..N)). Then composite those elements as matrix C.