The Algorithm for map reduce basically makes the following assumption -

Assumptions:

1. The input file which is a Text file is assumed to be of such a format
that when it is input to the mapper, the key is the offset or the line number in the file which is a longWritable and the value is the String present in the line.
2. It is assumed that only characters a-z, A-Z, whitespaces and 0-9 are considered in the text while processing the input.
3. The duplicates are considered in this process. For this purpose, it is assumed that as the iterable is accommodated in the main memory while reduce phase, the set containing unique words in the iterable value will also be accommodated.

Algorithm Logic-

Input for the Mapper -
LongWritable key and Text as the corresponding value
Output for the Mapper -
Key is in TextFormat and corresponds to the sorted words in the value text string. It can be concluded that there would be as many writes in the mapper function corresponding to the number of words in the text line.
Value is in TextFormat and indicates a list which is comprised of all the words composed of the same letters as that of the key.

The mapper takes the keys and values follows the following logic -
1. The key is emitted after sorting the characters present in the word.

- This is done by maintaining a map of 255 ascii characters.
- This map is incremented at the index of the character c where c is the character present in the word. ex. For a word “tat” - map[‘t’] = 2 and map[‘a’] =1.
- The map is iterated over again and characters are appended in ascending order thus getting a sorted string of the word in consideration.
- This sorted string is emitted as key and value is the word itself.

2. The Sort/Merge phase will sort and merge all the keys and its values as a list of words. ex. Words “top” and “pot” will end up under the key “opt”.

The reducer takes the values and keys emitted by the mapper and follows the following logic -
1. The key in the reducer is ignored.
2. The reducer just iterates over the Iterable<Text> list against each key and concatenates them with a delimiter ‘,’ and emits it as a key and null value.
3. A set is maintained to ensure that the values emitted are unique.