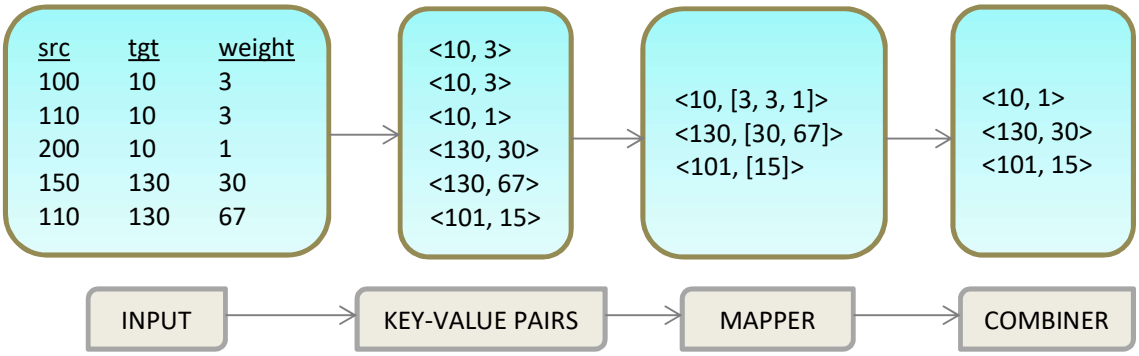
1. (a) Map-Reduce function as implemented:



In this function, the input data is a tsv file with three fields: source, target and weight. This data is pre-processed to split each line into a list of strings. The first string can be discarded since it isn't relevant. The second string serves as the key while the third is treated as value. After mapping, we generate key-value pairs representing the relation between the target and its inbound weight. These pairs are then reduced by key. For some key-value pairs, we iterate through the values to find the minimum value which is saved as the corresponding value for the key. This result is then saved into the system in the required output format as a tsv file.

1. (b) With reference to the idea discussed on http://codingjunkie.net/mapreduce-reduce-joins/, we design a MapReduce algorithm that joins record pairs from an unordered collection. The algorithm is as discussed below. In the mapper function, sorting only occurs by keys and the order of values is unknown. Hence we can use secondary sorting to tag keys with either "1" or "2" so that student name will always precede department name. Department_ID is set as key for both, Student and Department while a combination of title (i.e. 'Student' or 'Department') and name as value. In the combiner, data is combined by key using joinKey(Department_ID). A reduce function does not typically guarantee that its data is sorted. We therefore sort the output of reducer in ascending order of Department_ID before processing data into required format.

Pseudocode:

A. Generate key-value pairs

```
for line in file:
  if (line[0] == "Student")
       emit <line[2], line[0] + "-" +
line[1]>
  else
       emit <line[1], line[0] + "-" +
line[2]>
```

B. Mapper(file)

```
for value in values:
   if value.startWith("Department")
      Department_Name=value.split('-')[1]
   if value.startWith("Student")
      emit <key, value.split('-')[1],
Department_Name>
```

C. Reducer(key, values)

```
for value in values:
   emit in the format <Department_ID, Name, Department_Name>
sort(<key-values>) based on Department_ID
```

Pictorial representation: