Part A:

MapReduce can be divided into two process, which named mapper and reducer. Their inputs are both key-value pair, and outputs are both context, which is inner format in Hadoop. The specific process is illustrated using toy example as following.

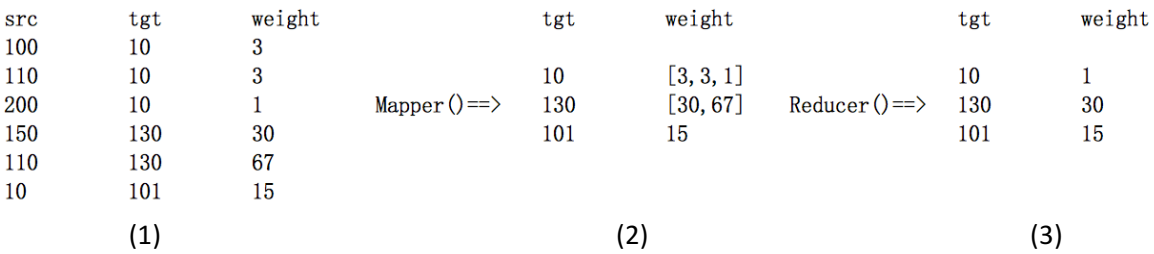

Figure 1. MapReduce(a)

As for the Mapper, there is a set of “key-value” pair from reading the tsv file. The “key” is the index of each line, which means the lines in tsv file. The “value” is the content of each line. In this case, we should split the value using tab and read the important information. Then, select the key variable and the value variable with the same key variable. The value variables are aggregated together. Finally, write “key” and “value” into context, which could be used for Reducer. In the toy example, we split the “value” with tab and derive the information of three dimensions. We aim at the smallest weight of one target, hence we can put the “src” aside and read two dimensions named “tgt” and “weight”. Because we designed to use “tgt” as the “key” of Reducer, we combine the “weight” to the same “tgt”, like process of the state (1) to state (2). Finally, we have the context like state (2) shows, which is ready to be used in Reducer.

As the Reducer, the source context is also “key-value” pair. The “key” is number of “tgt”, and the “value” is a number set of “weight”. We aim to pick out the minimum number of the set. Using the Iterator to scan the value once, we can pick the minimum number of “weight” set with the same “key”. As a result, write the “key” and the minimum number into context. From the example, when the “tgt” is equal to 10, we scan the number set of [3,3,1] and pick the 1 as minimum number, like the state (3) showing. Therefore, we can write the 10 and 1 into context.

Part B:

Because the two different information have the department ID as the same attribution, we use the department ID as the join key. When it comes to value, we can use some hints to represent their type. In this example, we can append the “Student” or “Department” to distinguish the data type. To be specific, the key-value pair can be represented like (Department_ID, “Student <student_name>” or “Department <department_name>”).

As for the Reducer, firstly we should split the value with space. Using “if” to judge the value type, and append different content with different hint. As a result, we can write the new “value” into context. The pseudo-code is as following.

Pseudo-code:

```
Mapper(key, value, context){
    String[] data = value.split(",");
    int id = 0;
    If(data[0].equals(Student)){
        id = Integer.parseInt(data[2]);
        Context.write(id, data[0]+" "+data[1]);
    }
    else{
        id = Integer.parseInt(data[1]);
        Context.write(id, data[0]+" "+data[2]);
    }
}
```

Figure 2. Mapper

```
Reducer(key, value, context){
    String dep = null;
    for value1 in value{
        String[] data = value1.split(" ");
        If(data[0].equals(Department)){
            dep = data[1];
            break;
        }
    }

    for value1 in value{
        String[] data = value1.split(" ");
        If(data[0].equals(Student)){
            Context.write(key, data[1]+" "+dep);
        }
    }
}
```

Figure 3. Reducer

Corresponding to illustration above, we use the example to make it more clear. From the original data, we select the Department_ID as the key, like 1234 and 1123. Then construct the value using the format "type + name". In this case, we can implement either "Student" or "Department" as the type. Hence, it can represent the data type and be a hint for Reducer to distinguish. Add the "student_name" or "department_name" into the value with the type. As a result, we write the value into context. The specific process shows as following.

```
Student      Alice    1234
Student      Bob      1234
Department   1123     CSE        Mapper()==>
Department   1234     CS
student      Carol    1123

                      key        value
                      1234       [Student Alice, Student
                                 Bob, Department CS]
                      1123       [Department CSE, Student
                                 Carol]

                 key       value
                 1123      Carol, CSE
Reducer()==>     1234      Alice, CS
                 1234      Bob, CS
```

Figure 4. MapReduce(b)