(1) [50 pts] Implement the following SQL statement using the MapReduce framework. Assume that there are two tables: student(sid, firstname, lastname, year, deptno, gpa) and dept(deptno, dname, campus, building) as follows. **Implement and explain your map and reduce function with execution snapshots.**

sid	firsname	lastname	year	deptno	gpa
10092	haley	lee	4	524	4.5
16283	markus	reed	3	514	3.1
17582	kingston	flores	3	555	4
17629	claude	baird	3	514	3.6
19460	julio	castillo	4	524	3.7
31345	will	smith	1	514	3.6
32245	eric	adams	1	534	2.6
33145	john	jones	1	524	3.2
42516	scarlet	wallace	2	514	4.3
47296	lucia	carty	3	534	4.2
56203	haley	lee	4	566	3.3
66392	leon	irving	3	514	2.9
75629	alice	parker	2	524	4
98562	kit	pierce	2	561	4.3

TABLE "student"

deptno	dname	campus	building
514	computer scienc	suwon	engineering
524	mathematics	suwon	natural science
534	biology	suwon	natural science
561	philosophy	seoul	humanities
566	economics	seoul	social science

TABLE "dept"

```
SELECT d.dname, max(s.gpa), d.campus
FROM student s
JOIN dept d ON s.deptno = d.deptno
GROUP BY d.dname
HAVING avg(s.gpa) > 3.5;
```

NOTE 1: You should write your codes using Hadoop streaming with Python.

NOTE 2: Your code should consist of 'map.py' and 'reduce.py.'

NOTE 3: You should use the following files (i.e., dept, student) as input.

Answer: **(Submit your code to i-campus. You don't have to write your code in the documentation.)**

Map function:

Column의 개수를 기준으로 student의 record인지, dept의 record인지 구분합니다. Column이 6개라면 student의 record로 간주합니다. Student의 record에서는 deptno를 key로 설정하고, gpa만을 value로 설

정합니다. Dept의 record에서는 detpno을 key로 설정하고, dname과 campus를 value로 설정합니다. 또한, 각 record가 어떤 table에 속했던 record 인지도, value 포함시켜줍니다.

Reduce function:

Reduce에서는 key값을 기준으로 group되어 입력을 받게 됩니다. Key인 deptno가 같은 record를 입력 받을 동안, value의 table 정보를 확인하여, student의 record인지, dept의 record인지 확인합니다. List를 만들어서, 같은 dept에 속한 학생들의 gpa를 저장하고, 같은 dept의 dname과 capus를 저장합니다. Key값이 다른 record를 입력 받을 때, 같은 deptno에 속했던 학생들의 평균 gpa를 구해서, 평균 gpa가 3.5보다 크다면, dname, 같은 deptno에 속한 학생들 중 최고 gpa, campus를 print하여, output 파일에 반환합니다. 마지막 key group에 대해서도 평균 gpa가 3.5보다 큰지 확인하고, 크다면 output 파일에 반환합니다.

-실행 화면

```
(base) taehyung@ubuntu:~/Desktop/DB/prob1$ $HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/share/hadoop/
tools/lib/hadoop-streaming-*.jar -input ./input -output ./output -mapper ./map1.py -reducer ./red
uce1.py
```

```
20/06/22 23:46:21 INFO mapred.LocalJobRunner: reduce task executor complete.
20/06/22 23:46:22 INFO mapreduce.Job:  map 100% reduce 100%
20/06/22 23:46:22 INFO mapreduce.Job: Job job_local1179968371_0001 completed successfully
20/06/22 23:46:22 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=409353
                FILE: Number of bytes written=1841026
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=19
                Map output records=19
                Map output bytes=381
                Map output materialized bytes=431
                Input split bytes=201
                Combine input records=0
                Combine output records=0
                Reduce input groups=6
                Reduce shuffle bytes=431
                Reduce input records=19
                Reduce output records=2
                Spilled Records=38
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=115
                Total committed heap usage (bytes)=695021568
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=660
        File Output Format Counters
                Bytes Written=61
20/06/22 23:46:22 INFO streaming.StreamJob: Output directory: ./output
```

-예제 문제의 output

```
(base) taehyung@ubuntu:~/Desktop/DB/prob1$ cat output/part-*
 mathematics,4.5, suwon
 philosophy,4.3, seoul
```

Dept

514, computer science, suwon, engineering
524, mathematics, suwon, natural science
534, biology, suwon, natural science
566, economics, seoul, social science
561, philosophy, seoul, humanities

Student

31345,will,smith,1,514,3.6
33145,john,jones,1,524,3.2
32245,eric,adams,1,534,2.6
42516,scarlet,wallace,2,514,4.3
75629,alice,parker,2,524,4.0
16283,markus,reed,3,514,3.1
66392,leon,irving,3,514,2.9
47296,lucia,carty,3,534,4.2
17629,claude,baird,3,514,3.6
19460,julio,castillo,4,524,3.7
10092,haley,lee,4,524,4.5
17582,kingston,flores,3,555,4.0
98562,kit,pierce,2,561,4.3
56203,haley,lee,4,566,3.3

(2) [100 pts] Solve the problem of finding a dominant set using the MapReduce framework. Assume that data is divided into multiple files. **Implement and explain your map and reduce functions with execution snapshots.**

[Definition 1: Dominant relationship]

For tuple $t_i \in R^n = \{t_{i1}, t_{i2}, \dots, t_{in}\}$, $t_i \prec t_j$ satisfies the following two conditions:

1) $\forall t_{ik} \in t_i, \forall t_{jk} \in t_j : t_{ik} \leq t_{jk}$

2) $\exists t_{ik} \in t_i, \exists t_{jk} \in t_j : t_{ix} < t_{jx}$

[Definition 2: Dominant tuple]

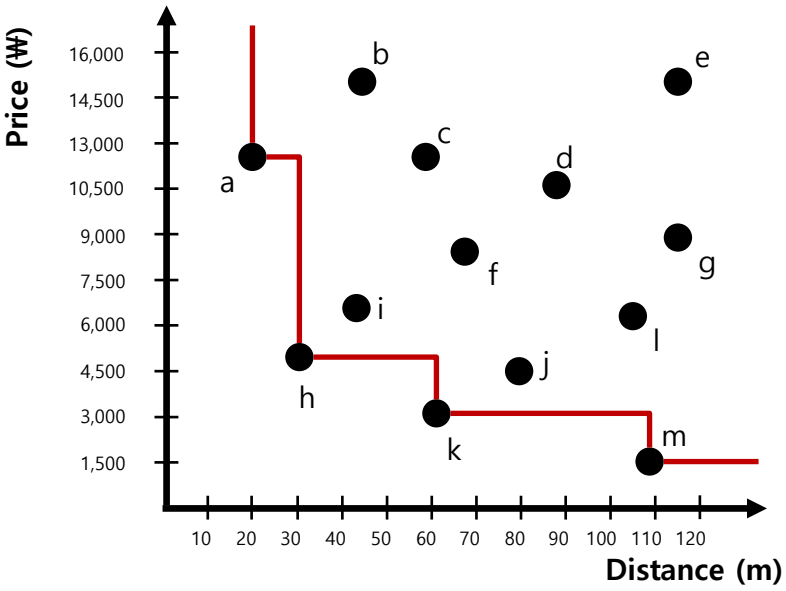
Given a set of tuples T, a set of dominant tuples is defined as:

$$Dom(T) = \{t \in T | \nexists\, p \in T \backslash t : p \prec t\}$$

[Example]

Name	Price (₩)	Distance (m)
a	12,500	20m
h	4,800	30m
i	7,000	43m
b	14,700	45m
c	12,500	58m
k	3,200	61m
f	8,500	67m
j	4,500	78m
d	10,700	88m
l	6,300	105m
m	1,600	107m
e	14,700	115m
g	8,800	115m

TABLE "Restaurant"



The dominant relationships for the relation “Restaurant” are:

$a \prec b,\ a \prec c,\ a \prec e,\ b \prec e,\ c \prec e,\ d \prec e,\ f \prec d,\ f \prec e,\ f \prec g,\ \ldots\ldots$

As a result, a set of dominant tuples is $Dom(T)$ = {a, h, k, m}

[Problem]

When recommending the restaurants according to two attributes, the distance and price, we want to find a set of dominant tuples. Implement an algorithm of finding a set of dominant tuples using the MapReduce framework.

NOTE 1: You should get the result in one MapReduce process.

NOTE 2: You should use the following files as input.

NOTE 3: You should write your codes using Hadoop streaming with Python.

NOTE 4: Your code should consist of ‘map.py’ and ‘reduce.py.’

Answer: (Submit your code to i-campus. Don’t write your code here.)

-map function: 각 파일들의 name, price, distance를 정보를 입력 받아서, 모든 record들에 대하여 같은 key값을 부여하고, value로는 name, price, distance를 부여합니다. 예를 들어,

Name :a , price: 12500, distance: 20 인 record에 대해서, key:1 , value: a, 12500, 20 으로 설정합니다. 다른 record들에 대해서도 동일하게 key값을 1로 설정합니다.

- reduce function: 모든 record들이 key 값이 1인 그룹으로 모두 묶이게 됩니다. 이 그룹 안에서, 각 record가 dominant tuple인지 확인합니다. 다른 record들과 비교하여, distance와 price가 모두 작은 record가 존재하지 않는다면, dominant tuple로 분류합니다. Dominant tuple로 분류된 record들의 name만을 output에 반환합니다.

- 실행화면

```
(base) taehyung@ubuntu:~/Desktop/DB/prob2$ $HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/share/hadoop/
tools/lib/hadoop-streaming-*.jar -input ./input -output ./output -mapper ./map2.py -reducer ./red
uce2.py
```

```
20/06/22 23:53:32 INFO mapreduce.Job:  map 100% reduce 100%
20/06/22 23:53:32 INFO mapreduce.Job: Job job_local83187869_0001 completed successfully
20/06/22 23:53:32 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=544743
                FILE: Number of bytes written=2435899
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=14
                Map output records=13
                Map output bytes=165
                Map output materialized bytes=209
                Input split bytes=318
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=209
                Reduce input records=13
                Reduce output records=4
                Spilled Records=26
                Shuffled Maps =3
                Failed Shuffles=0
                Merged Map outputs=3
                GC time elapsed (ms)=210
                Total committed heap usage (bytes)=912015360
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=140
        File Output Format Counters
                Bytes Written=24
20/06/22 23:53:32 INFO streaming.StreamJob: Output directory: ./output
```

-예제 문제에 대한 output

```
(base) taehyung@ubuntu:~/Desktop/DB/prob2$ cat output/part-*
k
m
h
a
```

[Input]: It is split into several files.

Restaurant1

a,12500,20
h,4800,30
i,7000,43
b,14700,45

Restaurant2

```
c,12500,58
k,3200,61
f,8500,67
j,4500,78
```

Restaurant3

```
d,10700,88
l,6300,105
m,1600,107
e,14700,115
g,8800,115
```

[Output]

Dominant_tuple

```
a
h
k
m
```