**MIS 6346 – Big Data Analytics**

Swetha Krishnakumar (sxk151530)

CREATE EXTERNAL TABLE twitter\_data(

retweet\_count int,

created\_at string,

text string,

id long,

source string,

in\_reply\_to\_screen\_name string,

User struct<location:string,

id:long,

id\_str:long,

name:string,

screen\_name:string,

geo\_enabled:string,

lang:string,

protected:string,

verified:string,

followers\_count:int,

friends\_count:int,

listed\_count:int,

favourites\_count:int,

statuses\_count:int,

profile\_background\_color:int>,

contributors string,

is\_quote\_status string,

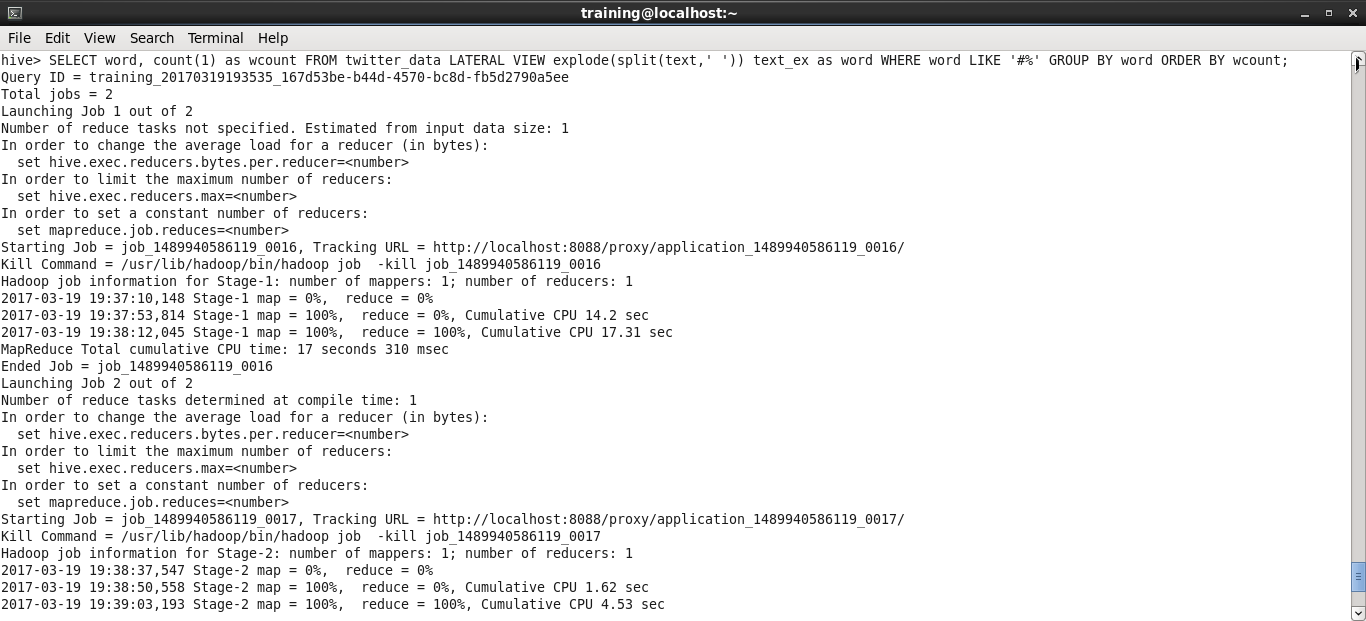
entities struct<user\_mentions:array<struct<screen\_name:string,name:string,id:long>>>)

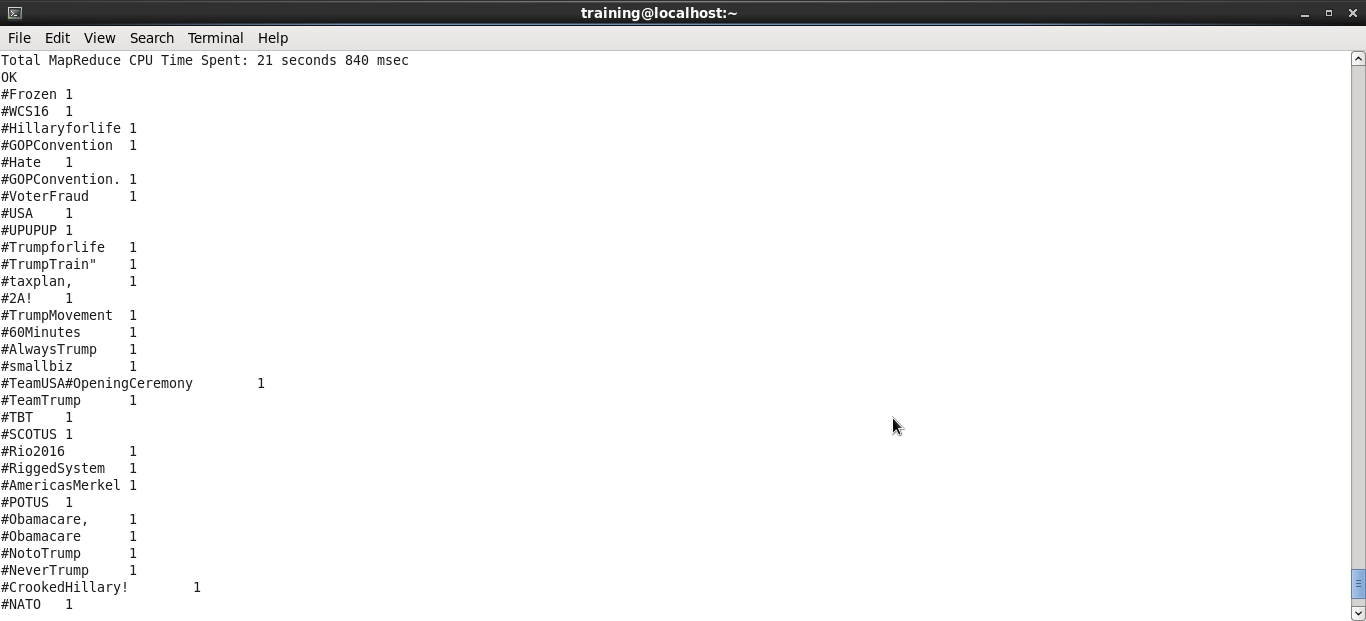
ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe';

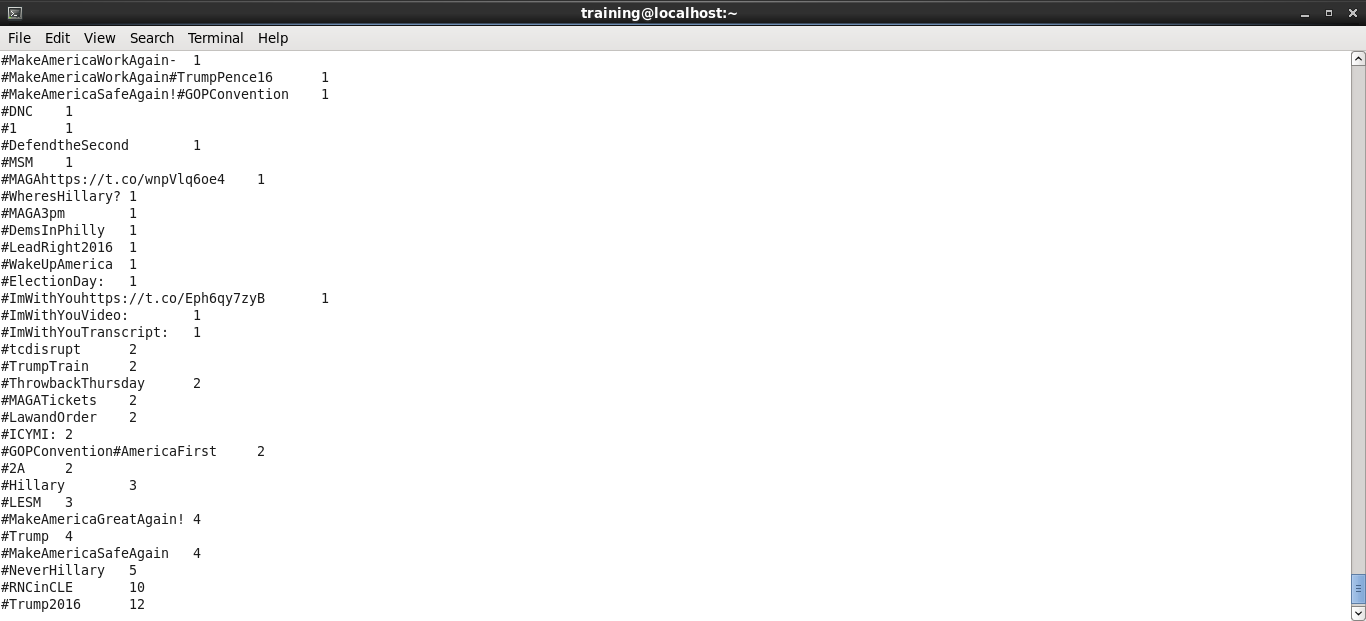
LOAD DATA LOCAL INPATH '/home/training/Downloads/Twitter.json' OVERWRITE INTO TABLE twitter\_data;

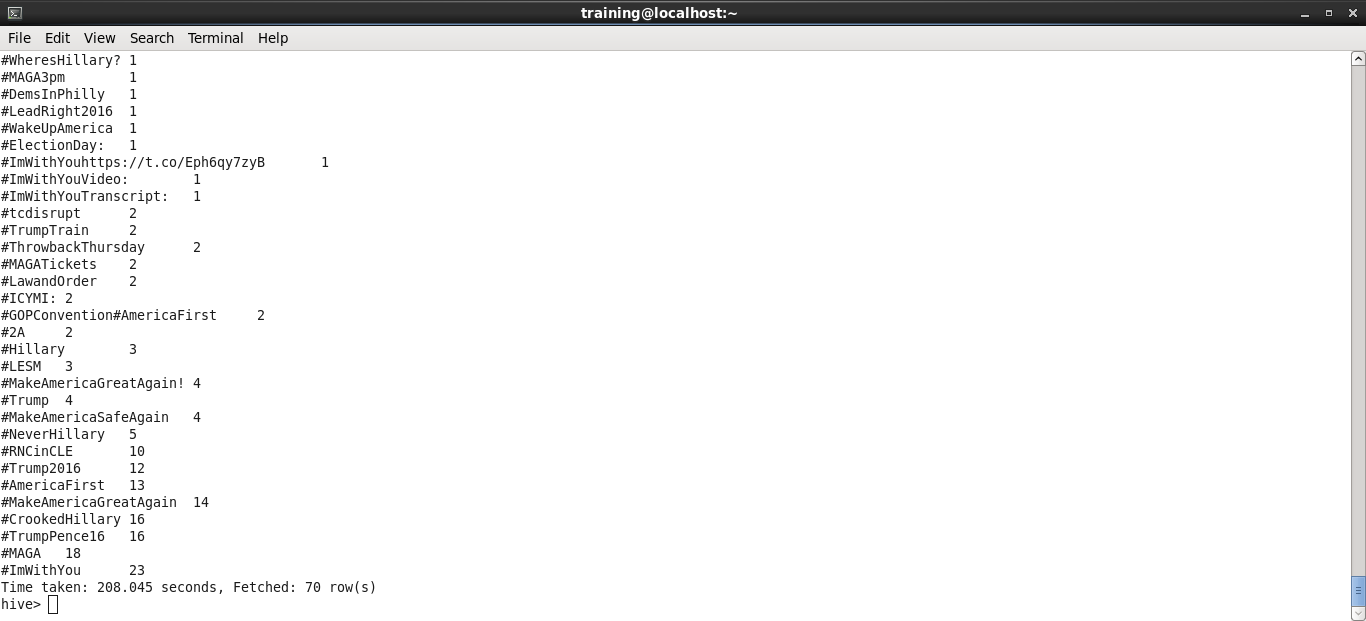
ADD jar /home/training/Downloads/hive-serdes-1.0-SNAPSHOT.jar;

1)a) SELECT word, count(1) as wcount FROM twitter\_data LATERAL VIEW explode(split(text,' ')) text\_ex as word WHERE word LIKE '#%' GROUP BY word ORDER BY wcount;

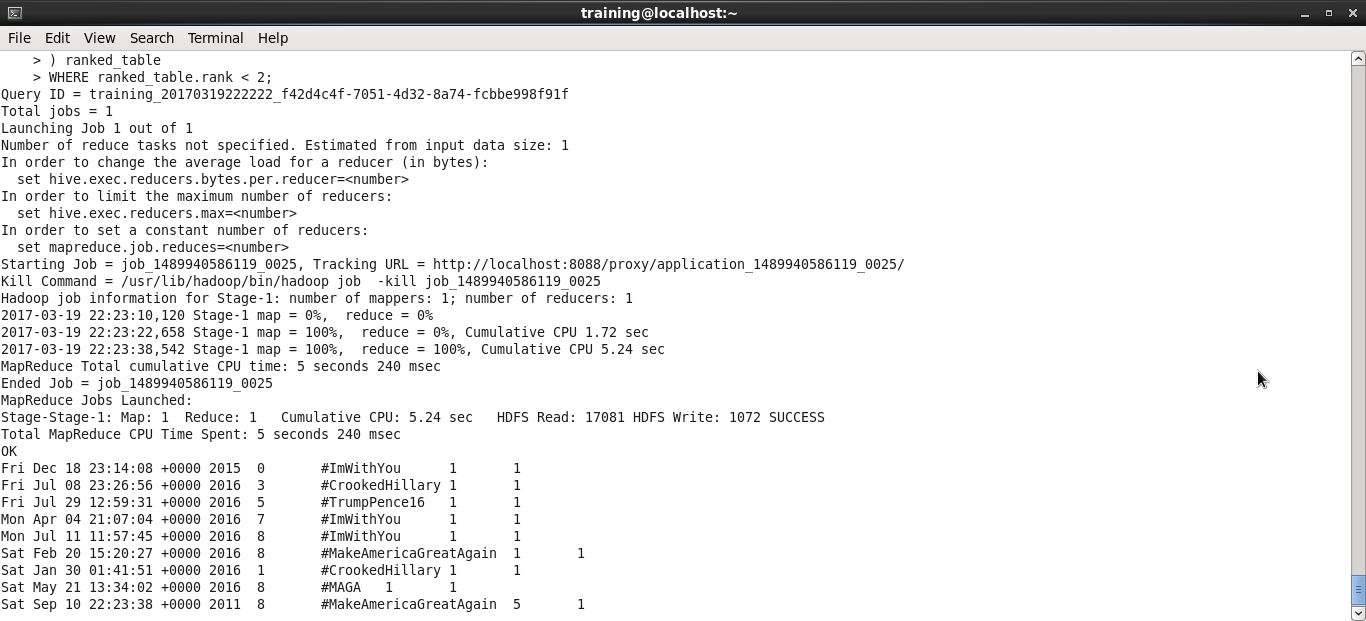


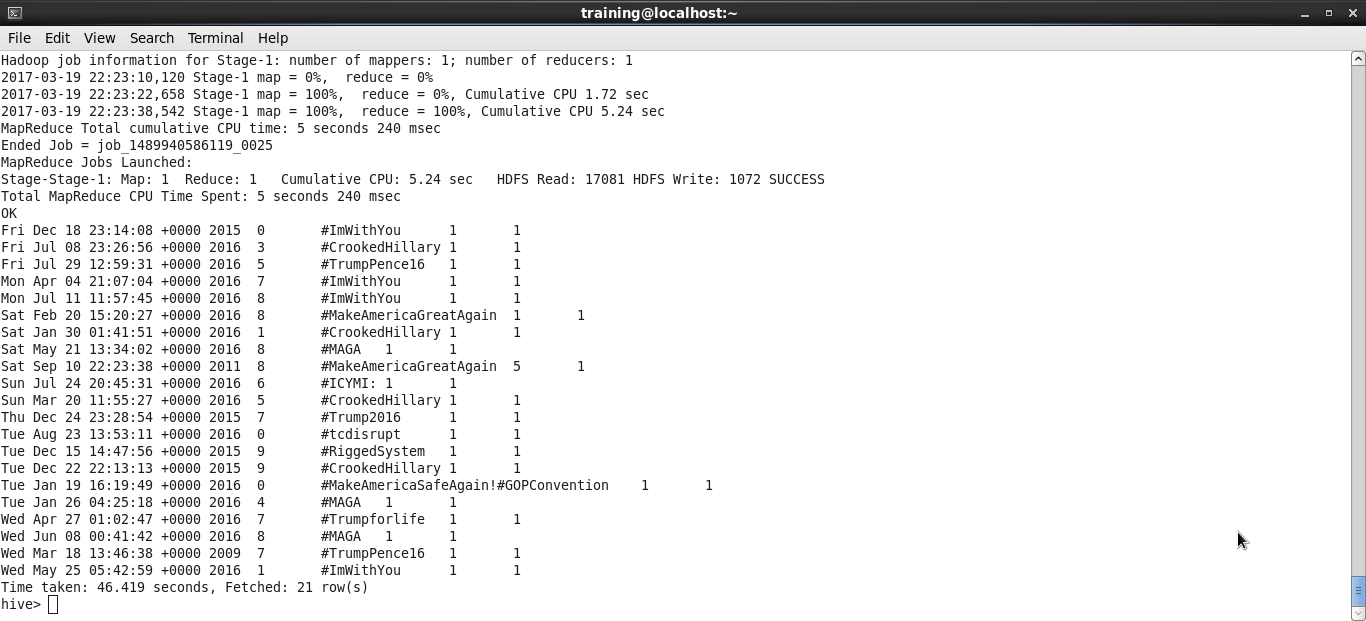




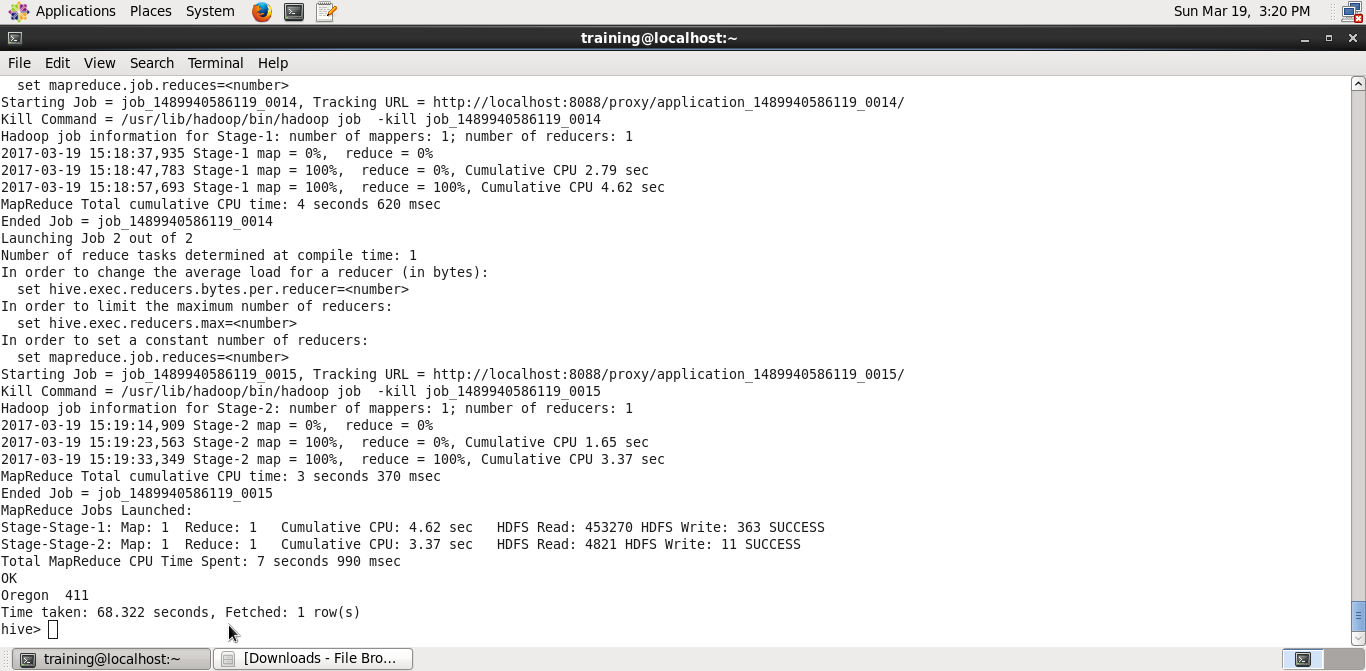


b) SELECT created\_at, retweet\_count, hashtag, count, rank  
FROM (  
    SELECT created\_at, retweet\_count, hashtag, count, row\_number()   
           over (PARTITION BY created\_at ORDER BY count DESC) as rank   
    FROM hashtags  
) ranked\_table  
WHERE ranked\_table.rank < 2;





c) select user.location, count(retweet\_count) as tweet from twitter\_data group by user.location order by tweet desc limit 1;



d) select user.name,user.followers\_count from twitter\_Data order by followers\_count desc limit 10;



e)

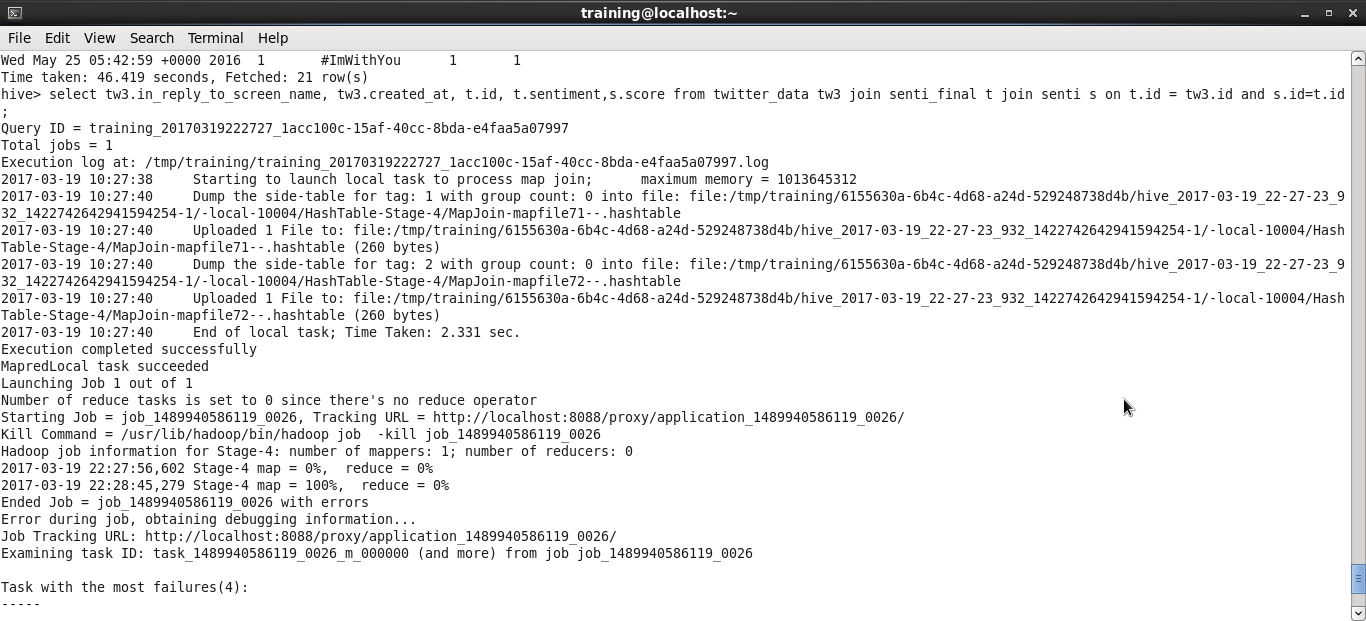
create table dict(word string,score int) row format delimited fields terminated by '\t';

LOAD DATA LOCAL INPATH '/home/training/Downloads/Dictionary.txt' OVERWRITE INTO TABLE dict;

create view l2 as select id, word from l1 lateral view explode( words ) dummy as word ;  
  
create view l3 as select  
    id,  
    l2.word,  
    case score  
      when  'negative' then -1  
      when 'positive' then 1  
      else 0 end as score  
 from l2 left outer join dict d on l2.word = d.word;  
  
select t.id,t.word, dict.score from l3 t join dict on t.word = dict.word;  
  
create table senti (id bigint, score int);  
  
insert into senti select t.id, dict.score from l3 t join dict on t.word = dict.word;

create view l1 as select id, words from twitter\_data lateral view explode(sentences(lower(text))) dummy as words;

select t.name, t.time, t.sentiment, senti.score from senti2 t join senti on senti.id = t.id;  
  
select t.name, t.time, t.sentiment, sum(senti.score) from senti2 t join senti on senti.id = t.id group by t.name, t.time, t.sentiment;



2) Yes. I faced issue in question e. Due to large data, the execution stopped after sometime.