作业 1（脚本基本操作练习）

给一天某社交网络数据，完成以下任务：

第一个任务：下载微博 1 天数据，地

址： http://101.236.63.184:12345/weibo/

这个目录下有两个文件： weibo.top10wan 是一个 10 万条数据记录的小文件。

weibo.all 是一个千万量级数据记录的大文件。

写实验代码可以在小文件上跑，最终在大文件上跑出结果。下载后用 7z 解压。

第二个任务：算出微博 1 天数据总量、微博发文 user_id 总量（uniq 的 user_id 的数量)，微博文章总量(uniq 的 weibo_id 的总量)

第三个任务：计算发文(weibo_id)最多的 top 100 个用户

作业 2（练习脚本并发）

通过并发的方式加速作业 1 中第三个任务的完成速度，用类似 map-reduce 的方式，第一步需要把一个大的任务切开，然后是计算，最后是 merge 结果，大致是这样。

因此这个计算发文最多的 100 个用户，大致需要如下过程。

1）用 split 命令将大文件切开，比如你有 4 个核，就先切 4 份。

2）分别用你之前的讨论得到每份文件中，发文最多的用户集合。

3）想想怎么 merge 在一起，再排序。

作业 3（学习 join 等脚本技巧）

热身：
文件 1（第一列数值不会重复，primary key）
0 a
1 b
3 c
文件 2（第一列数值不会重复，primary key）
0 e
1 f
2 g
希望能得到文件 3，如何实现？
0 a e
1 b f
2 g
3 c

提示：考虑用 join 命令，但是需要区分是第一列还是第二列，也可以不用 join 命令。
例如先处理成
文件 1.s
0 1 a
1 1 b
3 1 c

文件 2.s
0 2 e
1 2 f
2 2 g

再 join，在思考怎么做成文件 3. 不用 join 命令也可以。

任务： 之前的微博数据文件和长微博文件中，先统计，用户发文量列表{userid,weibo_cnt}，再统计用发长文量列表{userid，changweibo_cnt}，再归并出，用户发文和发长文总表{userid,weibo_cnt,changweibo_cnt}，并按照 userid 的发文数量做第一次降序排序，长微博文章量做第二次降序排序（sort 同时对两列进行排序，优先按第一列排序，第一列相同的情况下，按第二列顺序排）。

注：长微博数据也在这个目录下下载：http://101.236.63.184:12345/weibo/，本次集训的语料都在这个目录下下载。

作业 4（学习 grep 的使用）

热身：
读懂下面基本
mubiao.file 的内容如下
0 123 4
01 12 5
1 0 6
1 0 50
如果上面文件直接 grep 0 mubiao.file 则三列都会显示，如果只需要 grep 第一列第一个值是 0，怎么处理？

读懂并理解下面脚本 123.sh

```
value=$1
str=$\"^'${value}'\t'\'
cmd="grep "${str}" mubiao.file"
echo $cmd
eval $cmd
```

运行 sh 123.sh 0 即可看到结果。

任务 1：如何对 mubiao.file grep 第二列是 12 的结果
任务 2：如何对 mubiao.file grep 出第三列是 5 的结果
任务 3：有这样一个文件
梁斌 好人
学习 进步
人生 苦短
好好 学习
天天 向上
1）如何 grep 同时“学习”或者“好好”记录结果。(求包含“学习”和“好好”的并集)
2）如果需要匹配的关键词足够多，比如敏感词表，怎么用 grep 来做，敏感词表是按行写在一个文件中的，需要 grep 出所有包含敏感词的句子。
任务 4：grep 出当匹配关键词后的前一行和后一行，但不包含命中的这一行。
例如 grep 进步，结果返回
梁斌 好人
人生 苦短

作业 5

任务 1：统计微博小尾巴的用户总量和微博总量，排名按照用户总量倒叙排。微博小尾巴就是他发微博的设备名。

任务 2：统计用户发文设备数量排名，只用一个设备发文的用户不统计

输出的结果形如：

设备名	使用该设备发文的独立用户量	使用该设备发文的独立微博量
华为	180000	2000000
小米手机 2S	56000	30000000
…		
小米 6 拍人更美	2000	100
…		

第二个任务输出形如下面结果：

用户名	发文设备数
梁斌 penny	5
李开复	2
….	

注意：1）可能存在一个微博账号同一天用两种不同设备发微博

2）设备名中间可能有空格，例如“小米 5s 拍照黑科技”排序的时候，可能会造成错误

3）如何检测自己获取的数量是否正确？用 cut -f9,15 weibo.top10wan | grep "\tiPhone 客户端$" | cut -f1 | sort -u | wc -l 或类似的方法。

作业 6

任务 1：任意给定一个用户 id，找到这个用户 id 的所有发文，去重后，生成这个用户发文的词频表，分词可以用我的在线分词服务。http://api.pullword.com 或者自己的分词软件也可以。按照文档频率从高到低排列。

http://api.pullword.com/ 分词使用注意：

1. param1 可以过滤可能性较低的分词，例如：只想获得大于 0.5 可能性的分词，param1=0.5
2. param2 可以调整分词后是否显示可能性。当 param2=0 时，wget 出的结果可能会出现^M 字符
3. 当 source=文本中包含“#”符号时可能会导致错误，因此可将微博中的#符号去除。

这个脚本形如 sh gen_ci_ku.sh 梁斌 penny ，那么返回如下结果

关键词 文档频率

富人 0.1

苦大仇深 0.01

…

那么上诉结果的含义是，梁斌 penny 平均每 10 篇微博就要提一次富人，每 100 篇微博就要提一次苦大仇深。

注意：

1）文档频率的含义是这个文档中出现了就计算一次，而不是多次。

例如下面有 2 篇文档

A：“富人怎么可能苦大仇深，苦大仇深的是穷人”

B：“只有一种情况富人会痛苦，就是富人发现自己落后了”

C: “穷人苦啊”

那么在富人这个词出现在 2 篇文档中，文档集合总数是 3 篇，那么文档频率是 2/3 = 0.67；苦大愁深这个词虽然出现了 2 次，但是在同一篇文章中出现的，因此只记 1 次，最终这个词的文档频率是 1/3 = 0.33

任务 2：任意给一个词，判断这个词有那些用户喜欢用

这个脚本形如 sh gen_ci_ku2.sh 股市

关键词 文档频率

叶檀 0.1

梁斌 penny 0.01

…

这个含义就是股市这个词在叶檀的微博中，10 条提及 1 条，在梁斌 penny 的微博中 100 条提及 1 条，如果做广告，叶檀的效果更好。

作业 7

热身：
假定有两个大文件，每个大文件分别是一些关键词。例如：

文件 A：	文件 B：
中国	日本
日本	首都
首都	东京
北京	

现在求文件 C，文件 C 的内容是文件 A 和文件 B 都有的，例如上面这个结果 C 应该是：

文件 C：
日本
首都

作业内容：计算微博数据中的好基友。找到相互转发数量最大的基友 pair，按照相互转发量排序。

cut -f5,6,9,19,20,22 weibo.top10wan

这个命令可以得到转发人，转发人 id，转发的这条微博 id，被转人，被转人 id。

1650098112 亿毛-圈地自萌手帐 er 4240190337432738 GgRx7vbAS 3981267358 是一只废猪 SUE 4240077661747587

上面这个含义是，userid 为 1650098112 的用户（亿毛-圈地自萌手帐 er），转发了此前用户 3981267358（是一只废猪 SUE）发表的 weibo，这篇转文的微博 id 是 4240190337432738，被转文微博 id 是：4240077661747587,这个转文的 URL 可以通过 http://weibo.com/userid/url 得到。userid 就是 1650098112，url 就是 GgRx7vbAS，那么 URL 就是 http://weibo.com/1650098112/GgRx7vbAS

通过上述方法可以得到大量转发关系。现在要在这些转发关系中找到相互转发量最大的好基友。算法如下：
假定 A 转发了 B 一共 X 次（注意要对 weiboid 去重），B 转发了 A 一共 Y 次（注意要对 weiboid 去重）。那么 A 和 B 相互转发了 2*Min(X,Y)次。Min 的返回值是 X 和 Y 的最小值。
例如 A 和 B 之间发生了如下关系：

A 转 B weiboid：123
A 转 B weiboid：123
A 转 B weiboid：456
B 转 A weiboid：789
B 转 A weiboid：012
B 转 A weiboid：abc

那么 A 实际转了 B，一共 2 次，其中有一次抓取重复了，导致 weiboid 重复。B 转了 A 3 次。这样一来，A 和 B 相互转发了 2 * Min(2,3) = 4 次

作业 8（学习正则表达式）

热身阅读：

https://www.digitalocean.com/community/tutorials/using-grep-regular-expressions-to-search-for-text-patterns-in-linux

热身练习：

假定 test 是一个包含大量 ip 地址和其他乱七八糟内容的文件，请找出包括实际 ip 地址的行。

例如:

文件 test

192.168.1.1

0.0.0.0

Xxxxx

祖国伟大

使用如下包含正则表达式的方式，即可以挑出 IP 地址的文本行。

```
grep "[1-9][0-9]\{0,2\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}" test
```

或者

```
grep -E "[1-9][0-9]{0,2}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}" test
```

体会下上诉写法的区别。

作业：

1) 找到微博数据中，找到留下的包含 email 地址的文本行
   注意：邮箱中，@之前也可能包含“.”等特殊符号，@之后可能出现多个“.”
2) 找到微博数据的正文中（不是小尾巴中），找到提及小米品牌的文本行，并最终计算小米一共有多少个独立品牌，列出这些独立品牌，列出的多，说明正则召回率高。小米品牌包含红米 5A、小米 mix2s、小米 Note3、小米 7、红米 S2，小米 6X，小米 plus 7，等等，需要将所有后缀、空格都抓取到
3) 用 grep -E -o 来检测自己抓取到的结果是否完整。

作业 9（阅读作业）

1） 利用 timeout 命令的一些场合。

我们都知道 7z 是一个很好的压缩工具，这个压缩工具的压缩效率很高，适合存储大规模数据，但是 7z 压缩后的文件容易出现问题，例如压缩过程中主机关机，或者压缩过程中程序被错误的 kill 掉等各种情况，那么如果快速检测一个 7z 压缩后的文件是否是正确的呢？一般要用到 7za t filename。但如果文件很大，这个检测过程会非常长，如何快速检测呢？7za t filename 有个特点，如果运行后能持续下去一般文件就是正确压缩的，不必执行到结尾，这样通过一个 timeout 命令 让这个过程超时，从而快速返回，如果然后通过 shell 脚本监控进程的返回值，如果返回 2，说明文件坏了，这个 2 是 7z 程序的返回值，即还没有到 timeout 超时退出，程序就主动退出了，这种情况下返回 2；如果返回 124，这样说明程序是 ok 的，这种退出是 timeout 超时导致的退出，请大家仔细理解这个过程。

```
timeout 1 7za t $1 1>7za.t.log 2>7za.t.err
zat_result=$?

if [ $zat_result -eq 2 ]; then
        echo $1 is bad
fi

if [ $zat_result -eq 124 ]; then
        echo $1 is ok
fi
```

timeout 还有很多应用，例如 wget 一个网页，有时候可能时间特别长，那么就需要做一个 timeout 来规定最多的等待时间。那么 timeout 2s wget http://baidu.com 这种就比较常见。

2） 使用双括号（(())）进行四则运算

Shell 脚本做计算有很多方法，比较推荐双括号法。当然也可以用 let 和 expr。

例如：echo $((1+2*8+2))

还可以用这个做个简单的计算器，例如：

```
echo "shell 计算器"
read -p '输入一个数字: ' a
read -p '请输入计算方法: ' b
read -p '输入一个数字: ' c
echo "计算结果: "$(($a$b$c))
```

3） 运行时执行脚本

需要在 awk 里面执行一个脚本或者命令，参数是在 awk 一行行扫描的时候得到，例如下面这个真实场景中的例子。这个命令 list 出当前所有文件，print $9 得到这个目录中的所有文件名，然后在获得这些文件名的尾部 500 条记录。

```
ls -l * | tail -10 | awk -F" " '{print $9;}' | awk -F:"\t" '{ system("tail -500 " $1);}'
```

system 里面可以是命令，也可以是脚本，也可以是自己写的可执行程序，这样就大大扩展了 shell，awk 的执行能力。

4） shell 脚本中 read 命令的用法

比如文件 123 是

1　2

2　3

那么下面的命令将可以输出这个文件

```
cat 123 |
  while read field1 field2; do
    echo -e   $field1"\t"$field2
  done
```

read 命令可以从命令行也可以从管道读数据，大家可以感受下，之前的计算器是从 stdin 读数据。

5） 远程执行命令和脚本

例如下面这个命令，查询当前日期下某个文件的命令行数，而这个文件在主机 172.16.9.233 上，可以调用写如下脚本：

```
biz_mount=$(ssh 172.16.9.233 "wc -l /data/weixin/biz_log/weixin_biz.$CUR_DATE|cut -d\" \" -f1")
echo $biz_mount
```

也可以把需要在远程执行的命令写到脚本中，然后通过下面方式调用。

```
ssh 172.16.9.233 bash < xx.sh
```

xx.sh 中为需要执行的命令

有时候前者简洁方便，有时候后者方便。根据情况决定。这个在多机并行计算，计算结果后返回有重大意义。可以结合我们之前的 wait 命令一起，调动更多的机器一起计算，计算后在归并结果。

注：这个要能执行，需要之前做一个 ssh 免密码登陆主机 172.16.9.233 的操作，这个不再赘述。

最后，以上的一些也很难覆盖 shell，awk 的全部精华，但大多数实际工作中能遇到的，在做了这么多题目之后，大家会发现最常用的还是 sort 命令，sort 是全部大数据处理命令中的精华，大家一定要在大规模数据下进行操练，sort –T –S 这两个都经常用到-T 是临时文件存放处，这个存放内排序的临时文件，-S 这个设定的是内排序用的内存，可以用内存百分比，也可以指定内存大小，这些都是非常非常重要的，sort 命令在处理中文时容易跑偏，需要 export LC_ALL=”C” 确保排序正确。另外就是中文编码互转还会用到 iconv 这些都大家自己在日后工作中琢磨了，我们的基本作业就全部结束了，谢谢大家。

作业 10 微博地理位置信息大调研

给出 1 个月的微博签到数据，格式参见 readme，下载地址还是之前的老目录下。

数据中只有经纬度信息没有地理位置信息，需要通过百度 API 调用获得。调用方式见最后

任务：

1）最热门的国家签到排行（除中国）
签到国家\t uniq 的签到人数

2）最热门的中国城市排行
签到城市 \t uniq 的签到人数

3）最热门的中国省排名
签到的省 \t Uniq 的签到人数

4）微博用户签到城市数量排名和签到数量排名，join 到一张表，按照城市签到数量倒叙排列。(注意一个用户可能签到多次，但签到的城市是同样的一个)
微博 userid \t uniq 的签到城市总数 \t uniq 的签到总微博数

5）找到在 5 月份有跨国旅行的人，计算这部分人在微博总用户的比例。

6）跨国旅行的人喜欢用什么手机，首先需要找到有过跨国旅行的人，然后去 5 月 17 日那一天的微博数据中去兑出他们可能的手持设备，计算结果是一个手机排行榜，同时要求合并小米、苹果、华为、Oppo 和 Vivo，比如小米各种型号要归并成小米，苹果各种型号也要归并到苹果；其他杂牌手机不要求合并。
手机设备名 \t 出国次数

根据经纬度获取地理位置的 API：

1. 百度地理位置 API（只可调取国内省、市、街道信息）：
http://api.map.baidu.com/geocoder?callback=renderReverse&location=<LATITUDE>,<LONGITUDE>&output=json&pois=1
例如：
http://api.map.baidu.com/geocoder?callback=renderReverse&location=29.330403,113.146704&output=json&pois=1
2. Bing 地理位置 API（可调取国内外国家、省份，每天限制次数）：
http://dev.virtualearth.net/REST/v1/Locations/<LATITUDE>,<LONGITUDE>?o=xml&key=<YOUR_KEY>
申请 api key：
https://www.microsoft.com/en-us/maps/create-a-bing-maps-key
3. 无限制次数地理位置 API（可调取国内外国家、省份、城市）：
http://101.236.63.47:1234/gis?auth_user=freevip&latitude=<LATITUDE>&longtitude=<LONGITUDE>