Ex 2: Pig Latin

4 points
Weeks 5, 6 and 8

Start DFS, YARN, and Job History Server

```
> start-dfs.sh
> start-yarn.sh
> mr-jobhistory-daemon.sh start historyserver
```

Writing Your First Pig Program

Computing the degree of each node in the graph

Type "pig" in your command prompt to start grunt

Loading data from HDFS

```
grunt> A = load 'ex_data/roadnet/roadNet-CA.txt' as (nodeA:
chararray, nodeB:chararray);
```

Grouping

```
grunt> B = group A by nodeA;
```

Counting (group-wise)

```
grunt> C = foreach B generate COUNT(A) as freq, group;
```

Showing results

```
grunt> dump C
```

Your Task [Week 5]

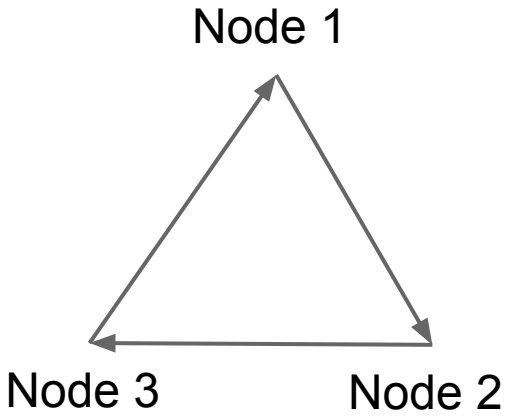
Extend node degree counting program to determine the following (submit graph.pig):

- **[E] (10%):** the frequency of each degree value
- **[E] (10%):** the percentage of dead-end nodes
- **[E] (10%):** the average degree of the graph

Another Program

Triangle Counting

Triangle in a directed graph



[E] (10 points): Write a Pig program with the following data flow

- Join two adjacent edges (n1,n2) and (n3,n4) on n2=n3 and create a “triad” relation (n1,n2,n4)
- Join the triad relation (n1,n2,n4) with the original edges (n5,n6) on conditions that n1=n6 and n4=n5 and create a relation called “triangle”
- Determine the size of the triangle relation

[D] (10 pints): Can you identify one problem with this approach here?

The Employee-Department Database

Download the data and copy them to your HDFS

```
> wget https://dl.dropboxusercontent.com/u/27408780/emp_dept.tar.gz
> tar xzf emp_dept.tar.gz
> hdfs dfs -put emp_dept
```

The Employee-Department Database

Use your favorite text editor to create a file called 'emp_dept.pig' and copy the following lines to the file

```
emp = load 'ex_data/emp_dept/emp.csv' as (empno:int, ename:
chararray, job:chararray, mgr:int, hiredate:datetime, sal:float,
deptno: int);


dept = load 'ex_data/emp_dept/dept.csv' as (deptno:int, dname:
chararray, loc: chararray);


salgrade = load 'ex_data/emp_dept/salgrade.csv' as (grade:int,
losal:int, hisal:int);
```

The Employee-Department Database

Checking the relations

```
dump emp;
dump dept;
dump salgrade;
```

If you get 14 rows from emp, 4 rows from dept, and 5 rows from salgrade then you are ready to move on.

Your Tasks [Week 6]

Express the following in SQL and Pig Latin (submit emp_dept.sql and one emp_dept.pig):

1. **[E] (4 points):** Smith's employment date
2. **[E] (4 points):** Ford's job title
3. **[E] (4 points):** The first employee (by the hiredate)
4. **[E] (4 points):** The number of employees in each department
5. **[E] (4 points):** The number of employees in each city
6. **[E] (4 points):** The average salary in each city
7. **[E] (4 points):** The highest paid employee in each department
8. **[D] (4 points):** Managers whose subordinates have at least one subordinate
9. **[D] (4 points):** The number of employees for each hiring year
10. **[D] (4 points):** The pay grade of each employee

Algebraic Interface

- An aggregate function takes a ***bag*** and ***returns a scalar value***
- Many aggregate functions can be computed ***incrementally***
- We call these functions algebraic
- COUNT is an example of an algebraic function
- The ***partial computations*** can be done by the ***map and combiner***
- The ***final result*** can be computed by the ***reducer***

Your Tasks [Week 8]

- [D] (5%): Create a function to compute the standard deviation of a group by implementing the algebraic interface:

```
1 public interface Algebraic{
2    public String getInitial();
3    public String getIntermed();
4    public String getFinal();
5 }
```

- [D] (5%): Describe the intermediate values that we need to maintain and how getFinal use them to compute the final result.