Final Project

DS730

In the final project you will be working with all of the technologies you have learned in this course. The only constraint for each of the problems is that you must solve them with one of the tools learned in this class: Java threads, Hadoop MapReduce, Pig, Hive or Spark. If you wish to solve them in some other fashion, you must ask me first.

You can download the files needed for the final project with the following command:
wget http://www.uwosh.edu/faculty_staff/krohne/ds730/final.tar.gz

If there is a tie for any of the questions, you should print out all of the correct answers. If you are using Hadoop, Pig, Hive or Spark, you must assume the data is stored in the HDFS folder of: **/home/ubuntu/final/Oshkosh/** for the Oshkosh csv file and **/home/ubuntu/final/IowaCity/** for the Iowa City csv file. If you are using the local filesystem to read in the files, you should not change the name of the files nor should you store them in a separate folder. For example, if you are writing a Java program to solve one of the questions, the input will be in the same folder as the Java file.

1. (21 pts) For the first problem, you will be reading in from several weather almanac files. You are encouraged to look at the files to see what the data looks like. Once you have a good grasp on what is contained in the files, you are to answer the following questions. You should note that all questions are not necessarily best solved using only 1 tool. It's possible that part (a) is best solved with Pig whereas part (b) is best solved with Java.
   Some of these problems may have a subjective meaning. If I ask which day was the hottest, that could be interpreted in different ways. Do I want to know which day had the highest temperature or do I want to know which day was the hottest with respect to the average? For example, assume one day was 94 the entire day and another day was in the 70's for most of the day and then jumped to 95 for a short amount of time before falling back to the 70's. Which day would be considered *hotter*? I will try and explain exactly what I am looking for. If a question feels ambiguous, ask me before attempting it.
   You should be aware of bogus values in the input. This is almost always the case in real datasets and you will almost always have to clean your data before you analyze it. For instance, some of the values are set to -9999 if no value was

recorded (or the cell is just empty). These values should not be included in your calculations. However, the rest of the row should not be discarded if an unimportant column has a bogus/missing value. For example, in the Oshkosh weather file, on January 26th, 2008, the temperature was recorded as -9999 but there is a valid wind speed. The temperature should be ignored but the wind speed should not be discarded. Make sure to look at the data before beginning so you can figure out what values are invalid. Ask me if you are not sure if a value is valid or not.

You are to solve each of these problems and provide the answers to all of the questions in a file called **output.txt**. For each of these problems, make sure that you create a separate script or program for each problem. For each problem, create an appropriately named file containing your solution. Your solution should be stored in *prob1W.XYZ* where W is equal to the actual problem letter and XYZ is the appropriate extension for the tool you are using. For example, if you solve 1a with java, your program will be stored in prob1a.java. If you solve 1b with pig, your program will be stored in prob1b.pig.

Your code does not need to output the exact answer for each of the questions (a-f) in problem 1 only. It is fine to do a little bit of extra manual work to find the answer. For example, for part (a), you do not need to output “cold is more common” or “hot is more common.” Rather, you could output the number of “cold” days along with the number of “hot” days and manually look at the output to determine the answer. However, the amount of manual work you do ought to be limited. An incorrect solution for number 1 would be to simply print out the entire weather file and then claim that you manually checked every row and counted the number of cold days and counted the number of hot days. If you are unsure if your solution does too much manual work, just ask.

All answers should be compiled into 1 file called output.txt.

a. In Oshkosh, which is more common: days where the temperature was really cold (-10 or lower) or days where the temperature was hot (95 or higher)?
b. When I moved from Wisconsin to Iowa for school, the summers and winters seemed similar but the spring and autumn seemed much more tolerable. For this problem, we will be using meteorological seasons:
   Winter - Dec, Jan, Feb
   Spring - Mar, Apr, May
   Summer - Jun, Jul, Aug
   Fall - Sep, Oct, Nov
   Compute the average temperature (sum all temperatures in the time period and divide by the number of readings) for each season for Oshkosh

and Iowa City. What is the difference in average temperatures for each season for Oshkosh vs Iowa City?

c. For Oshkosh, what 7 day period was the hottest? By hottest I mean, the average temperature of all readings from 12:00am on day 1 to 11:59pm on day 7.
d. Solve this problem for Oshkosh only. For each day in the input file (e.g. February 1, 2004, May 11, 2010, January 29, 2007), determine the coldest time for that day. The coldest time for any given day is defined as the hour that has the coldest average. For example, a day may have had two readings during the 4am hour, one at 4:15am and one at 4:45am. The temperatures may have been 10.5 and 15.3. The average for 4am is 12.9. The 5am hour for that day may have had two readings at 5:14am and 5:35am and those readings were 11.3 and 11.5. The average for 5am is 11.4. 5am is thus considered colder. Once you have determined the coldest hour for each day, return the hour that has the most occurrences of the coldest average.
e. Which city had a time period of 24 hours or less that saw the largest temperature difference? Report the city, the temperature difference and the minimum amount of time it took to obtain that difference. Do not only consider whole days for this problem. The largest temperature difference may have been from 3pm on a Tuesday to 3pm on a Wednesday. The largest temperature difference could have been from 11:07am on a Tuesday to 4:03am on a Wednesday. Or the largest difference could have been from 3:06pm on a Wednesday to 7:56pm on that same Wednesday. For a concrete example, consider Iowa City on January 1, 2000 at 2:53pm through January 2, 2000 at 2:53pm. The maximum temperature in that 24 hour span was 54 and the minimum temperature in that 24 hour span was 36. Therefore, in that 24 hour span, the largest temperature difference was 18 degrees. If this were the final answer, you would output “Iowa City”, “18 degrees” and January 2, 2000 3:53am to January 2, 2000 10:53am.
f. As a runner, I want to know when is the best time and place to run. For each month, provide the hour (e.g. 7am, 5pm, etc) and city that is the best time to run. The best time and place to run will be defined as the time where the temperature is as close to 50 as possible. For this problem, you are averaging all temperatures with the same city and same hour and checking how far that average is from 50 degrees. If there is a tie, a tiebreaker will be the least windy hour on average. If there is still a tie, both hours and cities are reported.

2. (14 pts) Due to budget cutbacks, the postal services at UW-Oshkosh can only afford 1 mail deliverer. Even worse, that deliverer is a student who works part-time. Postal services wants to minimize the amount of time that student has to work in order to save money. Because of this, they are interested in the fastest way to visit all buildings and return back to the Campus Services Building, BuildingOne in the example below. There are obvious routes that are terrible (e.g. going from one side of campus to the other and then back) but the optimal route is not obvious. Your goal is to read in a file that gives the time in seconds to get from a building to every other building and determine the best possible route such that you start at the building listed on the first line, visit all other buildings and end at the building listed on the first line. The building names in the example below are arbitrary and can be called anything. The input file you will read in is called **input2.txt** and will be formatted in the following manner:
BuildingOne : t(BuildingOne) t(BuildingTwo) t(BuildingThree) t(BuildingFour) t(BuildingFive)
BuildingTwo : t(BuildingOne) t(BuildingTwo) t(BuildingThree) t(BuildingFour) t(BuildingFive)
BuildingThree : t(BuildingOne) t(BuildingTwo) t(BuildingThree) t(BuildingFour) t(BuildingFive)
BuildingFour : t(BuildingOne) t(BuildingTwo) t(BuildingThree) t(BuildingFour) t(BuildingFive)
BuildingFive : t(BuildingOne) t(BuildingTwo) t(BuildingThree) t(BuildingFour) t(BuildingFive)

   Take the first line for example. t(BuildingTwo) will be an integer value denoting the number of seconds it takes to get from BuildingOne to BuildingTwo. On the first line, t(BuildingOne) will be 0 for obvious reasons. The input will always be formatted in this manner. If another building is constructed, it will be added to the end and the file will be updated accordingly. For example, if BuildingSix were constructed, the time to BuildingSix will be added at the end of every list and BuildingSix will be added to the end of the file. The time from BuildingOne to BuildingThree may not be the same as the time from BuildingThree to BuildingOne. There may be one way streets; it may be uphill, etc. A sample input file is shown below:

   BldgA : 0 5 6
   BldgB : 4 0 3
   BldgC : 6 4 0

Your goal is this, for the best route possible, print out the total time taken to start with BuildingOne, visit all buildings and then return to BuildingOne. You must also output the order in which you visited the buildings. You should save this output in a file called **output2.txt**.

3. (15 pts) The goal of the last problem is to answer questions about a large dataset that you are interested in. Amazon has many large datasets available at https://aws.amazon.com/public-datasets/. Here is another link to many large datasets: http://www.datasciencecentral.com/profiles/blogs/big-data-sets-available-for-free. You do not need to find all of your data from one place. You can combine data from multiple sources if need be. Find something that interests you. Once you have found a dataset that interests you, create 3-5 interesting questions about that dataset and answer them. The questions themselves are entirely up to you but they should be somewhat involved. Since the data might not be in a perfect format, you will likely have to clean and prepare your data before you can analyze it.
   For example, determining the most common wind direction in the weather file is quite trivial to do and would not be an acceptable question. A good example from the baseball document: one could try and come up with a query that would accurately predict the most valuable player for a given season. One could combine the weather and baseball datasets to determine the likelihood a player will hit a homerun given certain weather conditions (wind, temperature, humidity, location, etc). Because I do not know if everyone has taken the statistics course, I do not know your statistics background. Therefore, your use of statistics in your questions and answers is completely up to you. I also can't assume you have had the visualization course so feel free to submit your answer to this question in any reasonable format.
   If you can find a dataset that is many gigabytes, you'll be able to see the power of cloud computing on AWS. Assuming you found a dataset that is many gigabytes in size, transferring it from your local computer to S3 might take a long time. However, using wget on an EC2 server and then transferring it to S3 from your EC2 instance will be much faster. As a reminder from a previous activity, the following command uploads all contents of the current folder to some bucket in S3:
   aws s3 sync . s3://name-of-your-bucket-here/folderName
   To give you an idea of cost when running mapreduce on a large file, I've tested a file on the order of 10GB. I created 17 m3.xlarge instances (1 master and 16 core

nodes). With 17 instances, I ran a rather simple Pig script and it cost roughly $6. I ran a similar program in the past with 10 instances and noticed a considerable speedup. That particular running cost about $4. Choose your instance size wisely. This is not something you want to run more than a couple of times especially if you are using large instances.

The following are what you should submit, at minimum, for this problem:

a. Where you obtained your data. I do not need the entire dataset uploaded but a link to the data is sufficient.
b. The questions that you asked of your data in a reasonable format (i.e. a Word document).
c. The answers to the questions in a reasonable format.
d. The code you used to obtain your answers.

When you are finished, zip up your answers and upload it to the Final Project dropbox.