Lisa Chen
CS236-Project
Spring 2020

Summary

My approach was to follow the recommended breakdown of the tasks as it seems to be a good way to break down the problem to intermediate tables that will help with the search. I only did the requirements, so no extra credit was approached.

The runtimes are as follows (includes printing to console, so I/O for that was added – see explanation in the Start Up Work section). However, it doesn't include the final saving of the file in the runtime calculation.

Time to perform Task 1: 0.988s
Time to perform Task 2: 105.214s
Time to perform Task 3: 102.798s
Time to perform Task 4: 104.624s
Total Runtime: 313.624s or 5.227 minutes

The program takes 3 inputs as required: <locations csv folder> <recordings folder> <output folder>. It assumes that the data files are the same as what we are given (2006.txt, 2007.txt, etc).

After Task 4, a .csv file is created with the results (though it matches what is in the report) into a folder called "result csv files" into the output folder (Spark seems to have issues where it will not allow me to use an existing folder as the direct directory). There will be a few files generated, but the one that is the resulting csv will look something like this: part-00000-04cf59f3-096a-4b92-a4e9-3a3eb490f6e9-c000.csv.

To run the program, simply run the .bat file with the correct parameters. Here is an example:

```
➢  runproject.bat resources resources result
```

Doing this will run the .jar file with the parameters. For this example, all my data files are in the resources folder and I want to output to a results folder. If the wrong folder is pointed, it will throw an error through Spark itself; if the wrong number of parameters are specified, there is a custom message outputted to the console.

Start Up Work

To start my project, I got Spark Core library version 3.0.0-preview2 (as any version earlier than 3 would not work with Windows due to the Hadoop library that Spark relies on had issues working with my Java version and other issues). In addition, I also downloaded Spark SQL of the same version to work with Spark as the tutorials from Apache Spark's main Github generally uses Spark SQL. To fix some other issues I had, I also needed to change the Jackson core and databind library to version 2.10.4. All this was done using Maven to update the dependencies.

Finally, I also added the environmental variable HADOOP_HOME pointing to winutils.exe for Hadoop. This is required for Windows computers to run the program. For example, mine points to: C:\hadoop where my winutils.exe is in a bin folder.

First, I needed to convert all the data to DataSet objects for manipulation; the WeatherStationLocations.csv was an easy convert as Spark allows for easy convert of a .csv. For the recordings text files, I needed to create JavaRDD objects and StructFields to represent the schema that I wanted for this data. Since there were a lot of fields we didn't need, I removed them early at this step. In addition, I added a calculated precipitation column for the different case letters there were and a month column where it gets the string of the month based on DATEMODA.

System.currentTimeMillis() was used to calculate Runtime. Because runtime is hard to calculate given the asynchronous methods for creating the Dataset, start time was placed prior to any variable creations for the task and end time was placed after printing the task. It is assumed that the program will not print rows for values that are not done, so this was the only way I could figure out how to ensure that the DataSet was fully created. However, there is a bit of overhead for printing, which I am unable to avoid due to this method of runtime calculation.

Task 1

To perform task 1, which is to get all the stations in the US and grouped them by state, the first thing I did was filter the results from the DataSet representing WeatherStationsLocations.csv by CTRY = "US" and removed all the results where the State filed is null (since I am not approaching the extra credit and averaging all entries missing the state wouldn't give you any meaningful data). I then grouped them by state and put all the USAF IDs into one column as an array.

The result was 53 rows for 53 "states" as the data seems to include VI (Virgin Islands), DC (District of Columbia), and PR (Puerto Rico). I did not remove these as I felt it was nice to have even if they were not real states.

Please see below for the results for this task:

```
+-----+--------------------+
|State|               USAFs|
+-----+--------------------+
|   LA|[423630, 699560, ...|
|   NM|[690014, 696334, ...|
|   CA|[690020, 690070, ...|
|   UT|[690044, 690064, ...|
|   FL|[690090, 690524, ...|
|   MI|[690110, 690260, ...|
|   OK|[690130, 705022, ...|
|   NC|[690138, 690200, ...|
|   NV|[690170, 690174, ...|
|   TX|[690190, 690190, ...|
|   WA|[690230, 690240, ...|
|   AZ|[690310, 696454, ...|
|   OR|[690330, 720023, ...|
|   AR|[690500, 720156, ...|
|   AL|[691164, 720028, ...|
|   MD|[691174, 720012, ...|
|   AK|[691774, 692784, ...|
|   IN|[692644, 720161, ...|
|   GA|[692704, 720032, ...|
|   MS|[695354, 699744, ...|
|   VA|[695364, 699754, ...|
|   CO|[698414, 720009, ...|
|   IL|[699520, 720053, ...|
|   HI|[699530, 720025, ...|
|   SC|[699570, 720031, ...|
|   SD|[711680, 720008, ...|
|   MN|[720002, 720017, ...|
|   MA|[720006, 720034, ...|
|   OH|[720007, 720029, ...|
|   IA|[720013, 720054, ...|
|   NH|[720019, 720106, ...|
|   RI|[720033, 722151, ...|
|   ID|[720041, 720177, ...|
|   ME|[720047, 720128, ...|
|   KS|[720051, 720338, ...|
|   MT|[720058, 720073, ...|
|   PA|[720061, 720121, ...|
|   NY|[720065, 720178, ...|
|   WI|[720067, 720139, ...|
|   MO|[720076, 720169, ...|
|   CT|[720081, 720545, ...|
|   ND|[720082, 720491, ...|
|   NJ|[720116, 720179, ...|
|   WV|[720160, 720160, ...|
|   KY|[720163, 720353, ...|
|   TN|[720168, 720171, ...|
|   NE|[720308, 720308, ...|
|   WY|[720341, 720345, ...|
|   VT|[720492, 720493, ...|
|   DE|[720495, 720898, ...|
|   DC|[720618, 726402, ...|
|   PR|[785140, 785145, ...|
|   VI|    [785430, 785510]|
+-----+--------------------+

Time to perform Task 1: 0.988s
```

Task 2

To perform Task 2, I needed to manipulate the recordings Dataset. I did create a method to quickly create the Datasets separated for each recordings file, and eventually performed union on them to perform the task. Then, I needed to perform a join from Task 1 for the USAFs with the recordings' STN values to only have the recordings for the STN/USAFs that we care about. The union was performed with using Task 1 as the left DataSet since it is shorter than the recordings DataSet (Task 1 being only 53 states while the 2006 recordings text file alone has over 3 million lines). This would make it more efficient than if I were to do the join in the reverse direction.

After the join, I grouped each item by State and Month to perform average precipitation on the already calculated precipitation values (done at initialization of the DataSet), where the precipitation values were averaged when the state and month matched. Finally, I made a final DataSet where the month and average precipitation was aggregated into a single column and each state had a list of aggregated month/precipitation values (as an array). This was done using a special UDF (User Defined Function) in

Spark to combine the two column values together into a single value before I can use collect_list() to combine them into an array.

Please see below for the results for this task:

```
+-----+--------------------+
|State|         Precip List|
+-----+--------------------+
|   AZ|[November 0.00893...|
|   SC|[August 0.1457683...|
|   LA|[November 0.08694...|
|   MN|[November 0.01062...|
|   NJ|[March 0.07900149...|
|   DC|[January 0.0, Apr...|
|   OR|[October 0.054720...|
|   VA|[March 0.03249527...|
|   RI|[August 0.0610893...|
|   WY|[May 0.0636493374...|
|   KY|[December 0.16628...|
|   NH|[August 0.1361139...|
|   MI|[April 0.04466275...|
|   NV|[January 0.021177...|
|   WI|[April 0.06165948...|
|   ID|[April 0.04435056...|
|   CA|[December 0.06309...|
|   CT|[April 0.14679799...|
|   NE|[February 0.01015...|
|   MT|[May 0.0807792990...|
|   NC|[August 0.0804419...|
|   VT|[December 0.11962...|
|   MD|[July 0.055024326...|
|   DE|[October 0.103855...|
|   MO|[March 0.15046365...|
|   VI|[December 0.08630...|
|   IL|[May 0.0871694619...|
|   ME|[September 0.1328...|
|   WA|[April 0.06064643...|
|   ND|[January 0.014514...|
|   MS|[November 0.10504...|
|   AL|[March 0.12291313...|
|   IN|[May 0.1100024021...|
|   OH|[July 0.144883138...|
|   TN|[April 0.16501446...|
|   NM|[January 0.012061...|
|   IA|[May 0.0583075817...|
|   PA|[May 0.1073579339...|
|   SD|[May 0.0886068256...|
|   NY|[January 0.077787...|
|   TX|[May 0.0775800039...|
|   WV|[October 0.095235...|
|   GA|[July 0.111734122...|
|   MA|[August 0.1164332...|
|   KS|[October 0.116547...|
|   CO|[December 0.02631...|
|   FL|[February 0.09714...|
|   AK|[November 0.04172...|
|   AR|[June 0.131780525...|
|   OK|[December 0.03375...|
|   PR|[June 0.035441176...|
|   UT|[July 0.021499228...|
|   HI|[April 0.04544023...|
+-----+--------------------+

Time to perform Task 2: 105.214s
```

Task 3

To perform Task 3, I needed the DataSet I created for Task 2. The column with the month/average precipitation parsed using a special UDF where I looked at each month/precipitation value and found which was the lowest and highest month. Because I had to aggregate the month/precipitation into an array in Task 2, I had made them into Strings with a delimiter of a single space. Thus, I created an intermediate DataSet where I didn't parse the String and a DataSet using this DataSet object to split the month/precipitation column to corresponding columns for lowest month, lowest precipitation, highest month, and highest precipitation values.

Please see below for the results for this task:

```
+-----+------------+--------------------+-------------+---------------------+
|State|Lowest Month|Lowest Precipitation|Highest Month|Highest Precipitation|
+-----+------------+--------------------+-------------+---------------------+
|   AZ|    November|0.008935691105371218|         July|  0.07065813761059134|
|   SC|    November| 0.07339613086194678|       August|  0.14576830669591567|
|   LA|    November|  0.0869480834546337|         July|   0.1697039196631033|
|   MN|     January|0.003336326793773...|         June| 0.047394458223324575|
|   NJ|    February| 0.07410393700787403|         July|   0.1785957240038873|
|   DC|     January|                 0.0|      January|                  0.0|
|   OR|        July| 0.00966687020829092|     November|  0.14075152653828096|
|   VA|    February|0.024673968515683268|         June|  0.08022623892745996|
|   RI|    February| 0.05318513451892387|        April|  0.08669625871908687|
|   WY|    February|0.011197420634920623|          May|  0.06364933741080532|
|   KY|    November| 0.09035292213048407|         July|  0.16976356251813182|
|   NH|       March| 0.08385728710871762|         July|  0.23177604593929466|
|   MI|    February| 0.03219802952210654|    September|  0.07008190283855044|
|   NV|      August|0.009235637779941571|         July| 0.027827986142491327|
|   WI|     January|0.017938713700338508|         June|  0.08139457959880501|
|   ID|      August| 0.01603681242685391|     November|  0.06329878524462283|
|   CA|        July|0.002754643909795044|     February|  0.08149602463781755|
|   CT|       March| 0.07905631114292533|         June|   0.1607084785133567|
|   NE|     January|0.007690058479532156|       August|  0.10192715231788088|
|   MT|     January|0.010774703557312242|          May|  0.08077929906139672|
|   NC|     January|0.042857036651550114|         July|   0.1053681637641307|
|   VT|     January| 0.08598721023181453|         July|  0.24016064257028116|
|   MD|    February|0.036847265221878214|         June|  0.10497577917923413|
|   DE|    February| 0.04197385620915033|         June|  0.13578431372549024|
|   MO|     January| 0.06203433476394853|         June|  0.20050105600466106|
|   VI|       March| 0.04827044025157233|      October|  0.27268115942028986|
|   IL|     January|0.030057494726815015|          May|  0.08716946191474498|
|   ME|       March| 0.07291013950588417|         July|  0.16713952130764753|
|   WA|        July|0.014176592672802735|     November|  0.18510913163515094|
|   ND|    February|0.014421505376344075|         June|  0.10308791066095317|
|   MS|        June| 0.09098488888888889|     December|  0.17296009821976688|
|   AL|        June| 0.09880618965407002|         July|  0.18179667881733497|
|   IN|    November| 0.06111773801634633|         June|  0.12585707146426783|
|   OH|     January| 0.07794820143884895|         June|  0.15337783646322375|
|   TN|    February| 0.10388005301524189|     December|  0.18651829871414452|
|   NM|    November|0.006521143639787703|         July|  0.09166127292340888|
|   IA|     January|0.008268656716417905|         June|   0.0809310227658852|
|   PA|    February| 0.07852021478205941|         June|   0.1609032953639901|
|   SD|     January|0.011879822305354207|         June|  0.11713599231138885|
|   NY|    February| 0.07456230833565655|         July|  0.16723314606741563|
|   TX|    February|0.025009921011058464|         July|  0.09383419460271157|
|   WV|    February|0.062468800868149724|         June|  0.15884563532187407|
|   GA|    November| 0.05424255670503665|       August|  0.11492106723155303|
|   MA|       March| 0.10883711668657914|         July|  0.19126041666666663|
|   KS|     January|0.017501097935880543|         June|  0.19463163841807882|
|   CO|    February|  0.0115337224383917|       August|  0.07849496184984579|
|   FL|    November| 0.04829176678542824|         July|  0.26171101460718466|
|   AK|       March|0.028318786505874275|    September|  0.08177891420483052|
|   AR|    November| 0.07980119423955044|        April|   0.1912900976290096|
|   OK|     January|0.013254607264269103|         June|  0.10921756437367833|
|   PR|    February| 0.00864957264957265|          May|   0.2540740740740741|
|   UT|        July| 0.02149922890504515|      October|  0.05086682697812897|
|   HI|        June|0.025426452410383178|     December|  0.16028352919231498|
+-----+------------+--------------------+-------------+---------------------+
```

Time to perform Task 3: 102.798s

Task 4

Finally, to perform Task 4, I needed the DataSet created in Task 3. I simply use expr() and took the difference between the highest precipitation column and the lowest precipitation column to create the Difference column. In addition, there is a built-in sort that can sort the rows in ascending order without any manipulation.

Please see below for the results for this task:

```
+-----+------------+--------------------+-------------+---------------------+--------------------+
|State|Lowest Month|Lowest Precipitation|Highest Month|Highest Precipitation|          Difference|
+-----+------------+--------------------+-------------+---------------------+--------------------+
|   DC|     January|                 0.0|      January|                  0.0|                 0.0|
|   NV|      August|0.009235637779941571|         July| 0.027827986142491327|0.018592348362549756|
|   UT|        July| 0.02149922890504515|      October|  0.05086682697812897|0.029367598073083822|
|   RI|    February| 0.05318513451892387|        April|  0.08669625871908687|   0.033511124200163|
|   MI|    February| 0.03219802952210654|    September|  0.07008190283855044|  0.0378838733164439|
|   MN|     January|0.00333632679377...|         June| 0.047394458223324575| 0.04405813142955098|
|   ID|      August| 0.01603681242685391|     November|  0.06329878524462283| 0.04726197281776892|
|   WY|    February|0.011197420634920623|          May|  0.06364933741080532|  0.0524519167758847|
|   AK|       March|0.028318786505874275|    September|  0.08177891420483052| 0.05346012769895625|
|   VA|    February|0.024673968515683268|         June|  0.08022623892745996|  0.0555522704117767|
|   IL|     January|0.030057494726815015|          May|  0.08716946191474498| 0.05711196718792996|
|   GA|    November| 0.05424255670503665|       August|  0.11492106723155303| 0.06067851052651638|
|   AZ|    November|0.008935691105371218|         July|  0.07065813761059134|0.061722446505220116|
|   NC|     January|0.042857036651550114|         July|   0.1053681637641307| 0.06251112711258058|
|   WI|     January|0.017938713700338508|         June|  0.08139457959880501| 0.06345586589846651|
|   IN|    November| 0.06111773801634633|         June|  0.12585707146426783|  0.0647393334479215|
|   CO|    February|  0.0115337224383917|       August|  0.07849496184984579|  0.0669612394114541|
|   MD|    February|0.036847265221878214|         June|  0.10497577917923413| 0.06812851395735592|
|   TX|    February|0.025009921011058464|         July|  0.09383419460271157|  0.0688242735916531|
|   MT|     January|0.010774703557312242|          May|  0.08077929906139672| 0.07000459550408447|
|   SC|    November| 0.07339613086194678|       August|  0.14576830669591567| 0.07237217583396889|
|   IA|     January|0.008268656716417905|         June|   0.0809310227658852| 0.07266236604946728|
|   OH|     January| 0.07794820143884895|         June|  0.15337783646322375|  0.0754296350243748|
|   CA|        July|0.002754643909795044|     February|  0.08149602463781755|  0.0787413807280225|
|   KY|    November| 0.09035292213048407|         July|  0.16976356251813182| 0.07941064038764775|
|   CT|       March| 0.07905631114292533|         June|   0.1607084785133567| 0.08165216737043138|
|   MS|        June| 0.09098488888888889|     December|  0.17296009821976688| 0.08197520933087799|
|   PA|    February| 0.07852021478205941|         June|   0.1609032953639901| 0.08238308058193068|
|   MA|       March| 0.10883711668657914|         July|  0.19126041666666663| 0.08242329998008749|
|   TN|    February| 0.10388005301524189|     December|  0.18651829871414452| 0.08263824569890263|
|   LA|    November|  0.0869480834546337|         July|   0.1697039196631033|  0.0827558362084696|
|   AL|        June| 0.09880618965407002|         July|  0.18179667881733497| 0.08299048916326494|
|   NM|    November|0.006521143639787703|         July|  0.09166127292340888| 0.08514012928362118|
|   ND|    February|0.014421505376344075|         June|  0.10308791066095317| 0.08866640528460909|
|   NY|    February| 0.07456230833565655|         July|  0.16723314606741563| 0.09267083773175908|
|   DE|    February| 0.04197385620915033|         June|  0.13578431372549024|  0.0938104575163399|
|   ME|       March| 0.07291013950588417|         July|  0.16713952130764753| 0.09422938180176336|
|   NE|     January|0.007690058479532156|       August|  0.10192715231788088| 0.09423709383834873|
|   OK|     January|0.013254607264269103|         June|  0.10921756437367833| 0.09596295710940922|
|   WV|    February|0.062468800868149724|         June|  0.15884563532187407| 0.09637683445372434|
|   NJ|    February| 0.07410393700787403|         July|   0.1785957240038873| 0.10449178699601326|
|   SD|     January|0.011879822305354207|         June|  0.11713599231138885| 0.10525617000603464|
|   AR|    November| 0.07980119423955044|        April|   0.1912900976290096| 0.11148890338945915|
|   OR|        July| 0.00966687020829092|     November|  0.14075152653828096| 0.13108465632999006|
|   HI|        June|0.025426452410383178|     December|  0.16028352919231498|  0.1348570767819318|
|   MO|     January| 0.06203433476394853|         June|  0.20050105600466106| 0.13846672124071252|
|   NH|       March| 0.08385728710871762|         July|  0.23177604593929466| 0.14791875883057704|
|   VT|     January| 0.08598721023181453|         July|  0.24016064257028116| 0.15417343233846664|
|   WA|        July|0.014176592672802735|     November|  0.18510913163515094|  0.1709325389623482|
|   KS|     January|0.017501097935880543|         June|  0.19463163841807882| 0.17713054048219828|
|   FL|    November| 0.04829176678542824|         July|  0.26171101460718466| 0.21341924782175642|
|   VI|       March| 0.04827044025157233|      October|  0.27268115942028986| 0.22441071916871752|
|   PR|    February| 0.00864957264957265|          May|   0.2540740740740741| 0.24542450142450148|
+-----+------------+--------------------+-------------+---------------------+--------------------+

Time to perform Task 4: 104.624s
```