Rapport Final PLP

Bahri Akrem - Bouanen Mohamed

CentraleSupélec

2017/2018

Question 2.7 Displaying the content of a CSV file (Akrem Bahri)

Dans cet exercice, nous avons créé une classe Tree avec des attributs et des méthodes Static. Avec cette représentation, cette classe ne va pas être instanciée dans notre code Main. Les attribues de cette classe sont :

- *Year* : qui représente l'année de l'aber
- *Height* : qui représente la hauteur de l'arbre

La méthode implémentée dans cette classe est "*setparameters*" qui a partir d'une ligne du texte csv lue, renseigne les différentes caractéristiques de l'arbre.

Une autre méthode display est implémentée pour pouvoir afficher les caractéristiques de l'arbre dans une ligne de la forme :

Tree indice --- year : année valeur--- Height : hauteur valeur

Dans la fonction main, on parcoure les lignes du fichier csv, on initialise les paramètres de l'arbre selon la ligne lue et on affiche les caractéristiques de l'arbre.

- Input :

Le fichier arbres.csv doit être mis dans le dossier "Input" dans le répertorie du projet

- Résultat :

Voici un extrait du résultat obtenu :

Tree 1 --- year : 1935 --- Height : 13.0

Tree 2 --- year : 1854 --- Height : 20.0

Tree 3 --- year : 1862 --- Height : 22.0

Tree 4 --- year : 1906 --- Height : 16.0

Tree 5 --- year : 1784 --- Height : 30.0

Tree 6 --- year : 1860 --- Height : 45.0

Tree 7 --- year : 1840 --- Height : 40.0

Tree 8 --- year : 1933 --- Height : 16.0

Tree 9 --- year : #NA --- Height : 30.0

Tree 10 --- year : 1913 --- Height : 33.0

Tree 11 --- year : #NA --- Height : 30.0

Tree 12 --- year : #NA --- Height : 35.0

Tree 13 --- year : 1862 --- Height : 35.0

Question 2.8 Displaying the content of a compact file (Mohamed Bouanen)

De la même manière que l'exercice précédent, nous avons créé une classe *station* Static avec les mêmes méthodes (adaptées) et les attributs suivants : *Name, FIPS, altitude*.

La méthode *diplay* affiche les caractéristiques de la station de la manière suivante :

Station " + indice + " --- name : " + name + " --- FIPS : "+ FIPS + " --- altitude : "+ altitude

Dans la fonction main, on parcourt les lignes du fichier **isd-history.txt**, on initialise les paramètres de la station selon la ligne lue avec la méthode *setparameters* et on affiche les caractéristiques de station avec la méthode *display*.

- Input :

Le fichier input "isd-history.txt" doit être mis dans le dossier "Input" dans le répertorie du projet

- Résultat :

Voici un extrait du résultat obtenu :

Station 30062 --- name : GRAND FORKS AF --- FIPS : US --- altiture : +0277.7

Station 30063 --- name : HIBBING CHISHOLM-HIBBING AP --- FIPS : US --- altiture : +0413.6

Station 30064 --- name : LINCOLN 8 ENE --- FIPS : US --- altiture : +0362.4

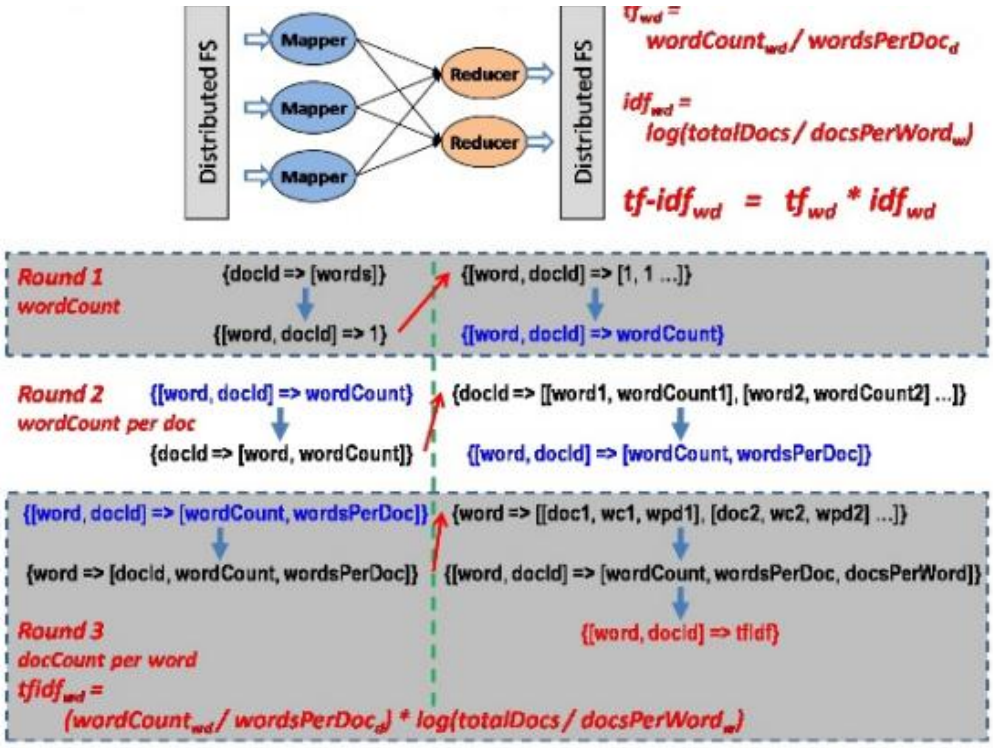
Station 30065 --- name : LINCOLN 11 SW --- FIPS : US --- altiture : +0418.2

Station 30066 --- name : TOK 70 SE --- FIPS : US --- altiture : +0609.6

Station 30067 --- name : RUBY 44 ESE --- FIPS : US --- altiture : +0078.9

Problem 1: TF-IDF (Akrem Bahri)

Le TF-IDF (de l'anglais *term frequency-inverse document frequency*) est une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus.



- <u>Méthode :</u>

Pour implémenter le calculateur de l'indicateur TF-IDF, on a eu recours à 4 jobs :

  - Job 1 sert à compter le nombre d'occurrence de chaque mot dans chaque document de l'input. L'output est de la forme (Word in Document, n)
  - Job 2 sert à calculer le nombre total de mots N dans chaque document de l'input pour enfin renvoie (Word in Document, n/N)

- Job 3 sert à calculer le nombre de document dans lequel chaque mot apparait et calculer finalement le TF-IDF. L'output est de la forme ; ((word in document), TFIDF)
- Job 4 sert à ordonner le résultat final selon le TD-IDF.

- Input et output :

Les fichiers input doivent être mis dans le dossier input

Les outputs sur chaque étape seront dans le dossier output

- Résultat :

word in Document	TF-IDF
the in THE CALL OF THE WILD.txt	0.07193926
the in ROBINSON CRUSOE.txt	0.04864976
and in THE CALL OF THE WILD.txt	0.04821259
i in ROBINSON CRUSOE.txt	0.04232389
and in ROBINSON CRUSOE.txt	0.03972252
to in ROBINSON CRUSOE.txt	0.03555207
of in ROBINSON CRUSOE.txt	0.02910232
of in THE CALL OF THE WILD.txt	0.02752294
he in THE CALL OF THE WILD.txt	0.02565644
was in THE CALL OF THE WILD.txt	0.02198671
to in THE CALL OF THE WILD.txt	0.021354
a in THE CALL OF THE WILD.txt	0.02068966
a in ROBINSON CRUSOE.txt	0.01864729
his in THE CALL OF THE WILD.txt	0.01774755
my in ROBINSON CRUSOE.txt	0.01765629
in in THE CALL OF THE WILD.txt	0.01695666
was in ROBINSON CRUSOE.txt	0.01664877
in in ROBINSON CRUSOE.txt	0.01598811
that in ROBINSON CRUSOE.txt	0.01555868
it in ROBINSON CRUSOE.txt	0.01522008

Question 5.2: Problem 2: Page Rank (Mohamed Bouanen)

Le *PageRank PR* est l'algorithme d'analyse des liens concourant au système de classement des pages Web utilisé par le moteur de recherche Google. Il mesure quantitativement la popularité d'un nœud d'un nœud d'un graphe.

- Méthode :

Pour implémenter un calculateur du Page Rank , nous avons procédé de la manière suivante :

Job 1 : sert à construire une liste de nœuds du graph et d'initialiser pour chaque nœud le PageRank. L'output est de la forme : <nœud> <page-rank> <link1>,<link2>,<link3>,<link4>,...,<linkN>

Job 2 : calcul le Page Rank et donne un output de la même forme que précédemment. Ce job peut être itérer plusieurs fois ce qui augmente la precision de l'algorithme. Dans notre implémentation, nous avons choisi 2 itérations.

Job 3 : récupère le PageRank et donne un résulte ordonne.

- Input and Output :

Les fichiers input doivent être mis dans le dossier input

Les outputs sur chaque étape seront dans le dossier Output

- Résultat :

user	Page Rank
18	47.07101058959961
4415	21.750492095947266
737	20.142515182495117
790	17.534427642822266
1753	17.32414436340332
143	17.248205184936523
1719	17.167720794677734
136	15.091687202453613
751	15.033056259155273
118	13.03643798828125

Problem 3: The trees of Paris

Contributions:

- **Nombre par type d'arbre : Mohamed Bouanen**
- **Hauteur maximale par type d'arbre : Akrem Bahri**

- Input and output :

Le fichier input doit être mis dans le dossier input

L'output sur le programme qui calcule le nombre d'arbres par type sera dans le dossier OuputTreeType

L'output sur le programme qui calcule la hauteur maximale d'arbres par type sera dans le dossier OuputTreeHeight

- Résultats :

Nombre d'arbres par type :

Type	Nombre d'arbres
Acer	3
Aesculus	3
Ailanthus	1
Alnus	1
Araucaria	1
Broussonetia	1
Calocedrus	1
Catalpa	1
Cedrus	4
Celtis	1
Corylus	3
Davidia	1
Diospyros	4
Eucommia	1
Fagus	8
Fraxinus	1
GENRE	1
Ginkgo	5
Gymnocladus	1
Juglans	1
Liriodendron	2
Maclura	1
Magnolia	1
Paulownia	1
Pinus	5

Platanus	19
Pterocarya	3
Quercus	4
Robinia	1
Sequoia	1
Sequoiadendron	5
Styphnolobium	1
Taxodium	3
Taxus	2
Tilia	1
Ulmus	1
Zelkova	4

- Hauteur maximale des arbres par type :

Type	Hauteur maximale
Acer	16
Aesculus	30
Ailanthus	35
Alnus	16
Araucaria	9
Broussonetia	12
Calocedrus	20
Catalpa	15
Cedrus	30
Celtis	16
Corylus	20
Davidia	12
Diospyros	14
Eucommia	12
Fagus	30
Fraxinus	30
GENRE	0
Ginkgo	33
Gymnocladus	10
Juglans	28
Liriodendron	35
Maclura	13
Magnolia	12
Paulownia	20
Pinus	30
Platanus	45
Pterocarya	30

Quercus	31
Robinia	11
Sequoia	30
Sequoiadendron	35
Styphnolobium	10
Taxodium	35
Taxus	13
Tilia	20
Ulmus	15
Zelkova	30

Annexe Eclipse Job flows console:

- TFIDF:

```
18/01/07 11:36:23 INFO mapreduce.Job: Job job_local1656652090_0004 completed successfully
18/01/07 11:36:23 INFO mapreduce.Job: Counters: 33
        File System Counters
                FILE: Number of bytes read=10468714
                FILE: Number of bytes written=11980864
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=11150
                Map output records=11150
                Map output bytes=491487
                Map output materialized bytes=513793
                Input split bytes=127
                Combine input records=0
                Combine output records=0
                Reduce input groups=404
                Reduce shuffle bytes=513793
                Reduce input records=11150
                Reduce output records=11150
                Spilled Records=22300
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=35
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=331235328
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=506577
        File Output Format Counters
                Bytes Written=495335
```

- Page Rank:

```
18/01/07 11:32:38 INFO mapreduce.Job: Job job_local97400069_0004 completed successfully
18/01/07 11:32:38 INFO mapreduce.Job: Counters: 33
        File System Counters
                FILE: Number of bytes read=217845978
                FILE: Number of bytes written=211783969
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=75880
                Map output records=75880
                Map output bytes=1051208
                Map output materialized bytes=1202974
                Input split bytes=132
                Combine input records=0
                Combine output records=0
                Reduce input groups=31936
                Reduce shuffle bytes=1202974
                Reduce input records=75880
                Reduce output records=75880
                Spilled Records=151760
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=64
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=496246784
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=4519716
        File Output Format Counters
                Bytes Written=1951855
```

- TreeType:

```
18/01/07 11:30:41 INFO mapreduce.Job: Job job_local887497174_0001 completed successfully
18/01/07 11:30:41 INFO mapreduce.Job: Counters: 33
        File System Counters
                FILE: Number of bytes read=36786
                FILE: Number of bytes written=530759
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=98
                Map output records=98
                Map output bytes=1233
                Map output materialized bytes=1435
                Input split bytes=108
                Combine input records=0
                Combine output records=0
                Reduce input groups=37
                Reduce shuffle bytes=1435
                Reduce input records=98
                Reduce output records=37
                Spilled Records=196
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=35
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=331227136
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=16778
        File Output Format Counters
                Bytes Written=410
```

- TreeHeight:

```
18/01/07 11:31:35 INFO mapreduce.Job: Job job_local792720995_0001 completed successfully
18/01/07 11:31:35 INFO mapreduce.Job: Counters: 33
        File System Counters
                FILE: Number of bytes read=36786
                FILE: Number of bytes written=530873
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=98
                Map output records=98
                Map output bytes=1233
                Map output materialized bytes=1435
                Input split bytes=108
                Combine input records=0
                Combine output records=0
                Reduce input groups=37
                Reduce shuffle bytes=1435
                Reduce input records=98
                Reduce output records=37
                Spilled Records=196
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=38
                CPU time spent (ms)=0
                Physical memory (bytes) snapshot=0
                Virtual memory (bytes) snapshot=0
                Total committed heap usage (bytes)=331227136
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=16778
        File Output Format Counters
                Bytes Written=444
```