

COMP9313 2018s2 Project 1 (25 marks)

Problem statement:

Build the inverted index for a given set of documents (compute the term weights by TF-IDF as shown in slide 5 of Chapter 4, using base 10 logarithm). Ignore the letter case, i.e., consider all words as lower case.

Input files:

Each line is in format of “DocID DOC”, where DocID is the ID of the document, and DOC is the document content (a list of terms). For example:

0 Construct Inverted Index
1 using MapReduce
2 inverted index code
3 index index file

This sample file “tiny-doc.txt” can be downloaded at:

<https://webcm>

[es/20465](https://webcm)

Output:

If a term W is contained in N documents, term contains N lines, in format of: “ W DocID₁, weight₁”, ..., and “ W DocID_n, weight_n”. tput for this

ascending order, and the term weights are of double precision (stored in double/DoubleWritable). Given the example graph, the output file is like (assumed in one file):

code	t2, 0.6020599913279624
construct	t0, 0.6020599913279624
file	t3, 0.6020599913279624
index	t0, 0.12493873660829993
index	t2, 0.12493873660829993
index	t3, 0.24987747321659987
inverted	t0, 0.3010299956639812
inverted	t2, 0.3010299956639812
mapreduce	t1, 0.6020599913279624
using	t1, 0.6020599913279624

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Code format:

Name your package as “comp9313.proj1” and name your driver class as “Project1.java”. Your program should receive 3 parameters: the input folder, the output folder and the number of reducers. Finally, package all your java files as a zip file with name “InvertedIndex.zip”.

Compile:

Your java code will be compiled and packaged as a jar file, and we will use the following commands to check the correctness of your solution:

```
$ $HADOOP_HOME/bin/hadoop jar YOURJAR.jar YOURCLASS input output 1
```

```
$ $HADOOP_HOME/bin/hdfs dfs -cat output/*
```

Please ensure that the code you submit can be compiled and packaged. Any solution that has compilation errors will receive no more than 5 points for the entire assignment.

Documentation and code readability

Your source code should be easy to understand. The ease of understanding your source code is

adability and

the codes) in your
ing scheme:

<https://eduassistpro.github.io/>

Add WeChat [edu_assist_pro](#)

Result correctness: 1
Efficiency: 5
Code structure, Readability, and Documentation: 2

Submission:

Deadline: Sunday 9th Sep 09:59:59 PM

Log in any CSE server (e.g., williams or wagner), and use the give command below to submit your solutions:

```
$ give cs9313 project1 InvertedIndex.zip
```

Or you can submit through:

<https://cgi.cse.unsw.edu.au/~give/Student/give.php>

If you submit your assignment more than once, the last submission will replace the previous one. To prove successful submission, please take a screenshot as assignment submission instructions show and keep it by

yourself. If you have any problems in submissions, please email to xuefeng.chen@student.unsw.edu.au.

Late submission penalty

10% reduction of your marks for the 1st day, 30% reduction/day for the following days.

Plagiarism:

The work you submit must be your own work. Submission of work partially or completely derived from any other person or jointly written with any other person is not permitted. The penalties for such an offence may include negative marks, automatic failure of the course and possibly other academic discipline. Assignment submissions will be examined manually.

Relevant scholarship authorities will be informed if students holding scholarships are involved in an incident of plagiarism or other misconduct.

Do not provide your assignment work to another person from whom it is submitted without your knowledge or consent. Do not provide or show work derived from it to another person from whom it is submitted without your knowledge or consent.