RMIT UNIVERSITY

School of Science

COSC2636/2632 Big Data Management

Assignment 1

	Assessment Type: Individual assignment; no group work. Submit online via Canvas→Assignments→Assignment 1. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements. This assessment supports CLO 1-5.
	Due date: 23:59pm, Sunday of Week 4; Deadlines will not be advanced but they may be extended. Please check Canvas→Syllabus or via Canvas→Assignments→Assignment 1 for the most up to date information. As this is a major assignment in which you demonstrate your understanding, a university standard late penalty of 10% per each working day applies for up to 5 working days late, unless special consideration has been granted.
	Weighting: 16 marks

1. Overview

Problem Description:
Given a set of items which achieve different scores based on different criteria, you need to find the top-k items which have the highest aggregated scores. The aggregated scores of an item A is the sum of all scores of A based on different criteria.

Requirement:
In the code skeleton (i.e., TopK.java), we have provided necessary APIs and implemented data preprocessing step. What you need to do: (1) understand the code skeleton and the function of each API; (2) implement the core function “thresholdAlgo()”. More specifically, you need to complete the missing part of the for loop in the function “thresholdAlgo()”.
Before you do the programming, please refer to the slide (i.e., “Topk Query.pdf”) for the pseudocode of the “thresholdAlgo()”.

Sample Test Case:
We provide two datasets for you to test your code. Each dataset has one million items with ten criteria. The sampled output “output.txt” is also contained in the assignment folder.

What you will learn:
You will learn how to use limited memory to efficiently find the top k items with the highest aggregated scores without scanning/loading the whole dataset.

If there are questions, you can email the tutors.

2. Assessment criteria

We will evaluate your program in terms of its correctness and efficiency on our test cases. We have prepared 10 test datasets each of which has the same size as the sample test datasets.

1. Correctness is worth 10 marks (i.e., one for each test), and efficiency is worth 5 marks.
2. We will not evaluate the efficiency of your program if your program fails to pass all test datasets. In this case, your marks cannot be greater than 9.

Note: The marking is based on the correctness of your implemented algorithm, as well as the efficiency upon the correctness of the results obtained.

1. For the efficiency evaluation, the marks is calculated based on the following conditions:
    1. X-Y< 1, efficiency marks=6.
    2. 1<X-Y< 2, efficiency marks=3.
    3. 2<X-Y< 3, efficiency marks=1.
    4. 3<X-Y, efficiency marks=0.

where X is the average running time (in seconds) of your program and Y is the one of our program. If your program is correct, you can easily get full marks.

2. No marks will be given for this assignment if you use brute force (a.k.a. scanning the whole dataset to get the top k items) to solve this problem, or do not use the code skeleton we provided.
3. No marks will be given if you change the function “main()” (e.g., output format and default parameter setting.).

3. Learning Outcomes

Please refer to the link (“http://www1.rmit.edu.au/courses/050436”) for the learning outcome.

4. Referencing guidelines

What: This is an individual assignment and all submitted contents must be your own (except for the main body of the code provided if any). If you have used sources of information other than that, you must give acknowledge the sources and give references using IEEE referencing style.

Where: Add a code comment near the work to be referenced and include the reference in the IEEE style.

How: To generate a valid IEEE style reference, please use the citethisforme tool if unfamiliar with this style. Add the detailed reference before any relevant code (within code comments).

5. Submission format

Submit **one .java file** showing the final output of your program via Canvas→Assignments→Assignment 1. It is the responsibility of the student to correctly submit their files. Please verify that your submission is correctly submitted by downloading what you have submitted to see if the files include the correct contents.

Note: One single java file: it contains your implementation of the threshold algorithm.

Naming convention: StudentNumber_A1.java (e.g., s1234567_A1.java)

6. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.
RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the University website.

7. Assessment declaration

When you submit work electronically, you agree to the assessment declaration.