COMP7106 Big data management [Section 2B, 2022]
Assignment 1
Implementation of query operators
Due Date: February 27, 2023, 5:00pm

Summary

The goal of this assignment is the development and testing of algorithms for some database operators (join, union, intersection, and difference).
You are going to use synthetically produced data which are given to you at Moodle. These data are in the form of text files and each text file can be seen as a relational table with two fields: A (string) and B (integer). Each line in a file corresponds to a tuple, where the values of A and B are tab-separated. For the records in file R.tsv the values of A are unique (i.e., A is a key for R). The same holds for file S.tsv. On the other hand, in file T.tsv, there can be more than one records having the same value in A. **Note that all files are sorted**. Open the files and make sure that you understand their contents.

1. Merge-join (40%)

Write a program, which reads files R.tsv and T.tsv, and computes and writes to a file RjoinT.tsv the natural join result of R and T, assuming that their common attribute is their first field (A). For example, the join between tuple (`'ab'`, `2`) from R.tsv with tuple (`'ab'`, `434`) from T.tsv should produce output tuple (`'ab'`, `2`, `434`). The output tuples of the join should be written to the output file separated by tabs; for example:

```
ab      2       434
ab      2       455
ab      2       918
...
```

Attention: Take advantage of the fact that the A values in R.tsv are unique to design and implement an efficient algorithm that requires only one pass over the files.
Your program should implement the merge-join algorithm described at the notes and not any variant of it. This means that:
1) The lines of files R_sorted.tsv and S_sorted.tsv should be read once only
2) You will not read the entire files in data structures in memory (e.g., arrays) before the join algorithm starts. For each line that you read from each file you should make sure that you find the results that correspond to that line.

Programs that violate the above guidelines will not receive full marks.

2. Union (20%)

Write a program that reads files R.tsv and S.tsv, and computes and writes to a file RunionS.tsv the union of R and S, assuming that the two relations have the same schema. Your program should read each line from R and S *just once* and compute the union at the same time. Again, you are not allowed to load all data in memory before computing the union. Example output:

```
ab   2
ad   3
ah   1
...
```

3. Intersection (20%)

Write a program that reads files R.tsv and S.tsv, and computes and writes to a file RintersectionS.tsv the intersection of R and S, assuming that the two relations have the same schema. Your program should read each line from R and S *just once* and compute the union at the same time. You are not allowed to load all data in memory before computing the intersection. Example output:

```
bk   5
ce   7
cq   6
...
```

4. Set difference (20%)

Write a program that reads files R.tsv and S.tsv, and computes and writes to a file RdifferenceS.tsv the difference R – S, assuming that the two relations have the same schema. Your program should read each line from R and S *just once* and compute the difference at the same time, implementing a variant of the merge-join algorithm. Again, you are not allowed to load all data in memory before computing the difference. Example output:

```
ab    2
ad    3
ah    1
...
```

Deliverables: You should submit your 4 programs and a single PDF file which documents the programs and any special instructions for compiling and running them. Please submit a single **ZIP** file with all requested programs and documents to Moodle on or before 5:00pm, February 27th, 2023. Make sure all contents are readable. **Please do not submit any data files**. Please feel free to post your questions on **Moodle forum** or contact the TA of the course if you encounter any difficulty in this assignment. We would be happy to help.