University of Windsor

School of Computer Science

COMP 8547 Advanced Computing Concepts – Winter 2026
Course instructor: Dr. Olena Syrotkina

Assignment 2

Objective: The aim of this assignment is to implement some components which can be used for the final project. Those include web page dictionary, ranking and spellchecking by using hash tables, heaps and search trees (please use Java and Eclipse or IntelliJ IDEA), etc.

Rules: **This is an individual assignment.** You are allowed to discuss problems and ideas within your group. However, please keep in mind that you are not allowed to share this assignment with other students from your Section or other Sections.

Instructions:

1. Please combine all the CSV files your group received when completing Assignment 1. If your file is in a different format, you can use that file instead.

2. Please complete **one** of the tasks listed below. Each student in a group must choose a different task, ensuring **no two students in the same group work on the same task**. Submit your report individually and **specify** which task you completed.

Task 1: Spell Checking

Objective: Implement a spell checker using hashing techniques and sorting algorithms.

Vocabulary:

A. Create a vocabulary from a set of csv/text files.
B. Use a hashing technique (e.g., cuckoo hashing) to store the vocabulary for efficient lookup.

Alternative Word Suggestions:

A. Implement an edit distance algorithm to suggest alternative words.
B. Use merge sort to sort the suggestions based on their edit distance.

Task 2: Word Completion

Objective: Implement a word completion feature using self-balancing search trees.

Vocabulary:

A. Create a vocabulary from a set of csv/text files.
B. Use an AVL tree or red-black tree to store the vocabulary (each node should store a word and its frequency of occurrence).

Autocomplete Functionality:

A. Implement a function that returns word completions based on a given prefix. Traverse the tree to find the node corresponding to the prefix and collect all words in the subtree.
B. Rank the word completions based on their frequency of occurrence. Use a min-heap to store the top suggestions and sort them before displaying to the user.

Task 3: Frequency Count

Objective: Count the frequency of words in each URL using sorting algorithms.

Word Frequency Calculation:

A. Parse the text from all CSV/txt files to extract words.
B. Use a hash table to count the occurrences of each word.

Frequency Sorting:

A. Use a sorting algorithm (e.g., quick sort or heap sort) to sort the words based on their frequency.
B. Display the top N most frequent words to the user.

Task 4: Search Frequency

Objective: Track and display the search frequency of words using self-balancing trees.

Search Query Collection:

A. Implement a function to handle search queries from users.
B. Use a self-balancing search tree (e.g., AVL tree, red-black tree) to store each search query and its frequency.

Search Frequency Update:

A. Each time a word is searched, update its count in the search tree.
B. Maintain a log of search queries and display top searches to the user.

Task 5: Page Ranking

Objective: Implement a page ranking system using heaps and sorting algorithms.

Content Parsing and Frequency Sorting:

A. Parse the content of web pages to extract keywords. Use a self-balancing search tree (e.g., AVL tree, red-black tree) to store the frequency of each keyword on each page.
B. Use sorting algorithms (e.g., quick sort or heap sort) to sort keywords based on their frequency within each page.

Page Ranking Calculation:

A. Calculate the rank of each page based on the frequency of search keywords.
B. Use a max-heap to keep track of the highest-ranked pages. Display the highest-ranked pages to the user.

Submission requirements:

1. You will earn a maximum of **100 points (accounts for 5.25%)** for successfully completing this assignment and submitting both your report and source code within the specified deadline.
2. You must submit:
   I. A report (in PDF or word), in which you provide the following elements:
   - Your task (e.g. Task 1 or Task 2 or Task 3 etc.).
   - Explanations for the solution provided (explain how you solved your task).
   - Outputs (screenshots) with comments and explanations (each screenshot must be numbered (e.g. Fig 1. Displaying the number of ….) and explained what we can see in your screenshot).

   II. All Java source code files and classes (**in both *.java and *.txt format**) needed to run your programs. Your source code must be well-commented. **Do not upload your source code to Brightspace as a single zip file. Such submissions will not be accepted.**

3. Marks will be **deducted** if comments/explanations are missing.

4. **This assignment is subject to a plagiarism check. The plagiarism check originality score must not exceed 50%. No points will be awarded for assignments submitted via email, Teams, or other platforms, for sending zip archives, or for failing to submit your code in *.txt files.**

5. Assignment submission after the deadline will receive a penalty of 10% for the first 24 hrs, and so on, for up to three days. After three days, the mark will be zero.

6. Unlimited resubmissions are allowed. But keep in mind that we will consider the last submission. That means that if you resubmit after the deadline, a penalty will be applied, even if you submitted an earlier version on time.

Academic Integrity.

Plagiarism is a serious offense and can result in severe consequences such as receiving a grade of **zero** on the assignment/lab or even being asked to leave the program.

Copying or using someone else's code is considered plagiarism. This includes using code from online sources, previous labs/assignments, or from other students. Even if you have modified the code, our antiplagiarism software can still show it as plagiarism.

To avoid plagiarism, make sure to always use your own words and ideas when writing source code. Additionally, always check with your instructor to make sure you understand what is allowed and what is not in terms of using outside resources.

Remember that academic integrity is essential for your own personal and professional growth, and it is your responsibility to uphold these principles. So please take it seriously and always produce your own original work.