**EVALUATING THE PERFORMANCE OF SEARCH AND SORT ALGORITHMS IN NEWS CLASSIFICATION USING ARRAYS AND LINKED LISTS**

*(Working Duration: Monday of Week 6 to Friday of Week 9 – 30 Marks)*

**Background:**

In recent years, the rapid growth of digital media has significantly transformed how we consume news and information. News is now accessible at the click of a button, with online platforms and social media enabling people to stay updated in real-time. However, this convenience has also led to a new set of challenges, particularly the spread of misinformation and "fake news." Fake news refers to deliberately fabricated stories or misleading information designed to deceive the audience, often for personal, political, or financial gain. These false narratives can spread quickly across the internet, particularly through social media platforms, where they can easily reach millions of users in a short amount of time.

The rise of fake news has serious implications for public trust, decision-making, and societal well-being. False or misleading information, especially when it pertains to sensitive topics such as health, science, technology, and environmental issues, can have dire consequences. For instance, during a health crisis, misleading claims about medical treatments or vaccines can cause panic, harm, and loss of life. Similarly, fabricated reports about technology or environmental issues can create confusion and misguide public action. Consequently, it is crucial to differentiate between true news and fake news to preserve the integrity of public discourse and decision-making processes.

To combat the spread of fake news, researchers and data scientists have turned to computational techniques, particularly search and sort algorithms, to help identify and filter reliable information. By analyzing large datasets of news articles, these algorithms can help detect patterns, flag potential misinformation, and prioritize trustworthy sources. Various search algorithms and more advanced sorting algorithms are commonly employed to process news data efficiently. These algorithms not only help identify key trends but also ensure that the most relevant and credible content is highlighted, reducing the influence of misleading information.

In this assignment, the goal is to evaluate how different search and sorting algorithms perform when processing news data, with a focus on distinguishing between true news and fake news. The assignment also aims to explore how the choice of data structure—such as Arrays or Linked Lists—affects the performance of these algorithms in terms of time and memory efficiency.

**Question:**

In this assignment, you are provided with two datasets: one containing true news articles (***true.csv***) and another containing fake news articles (***fake.csv***). Each article includes the title, content, subject categories, and publication date. Your task is to apply various search and sort algorithms to analyze both datasets and answer the following questions, but not limited to:

1. How can you efficiently sort the news articles by year and display the total number of articles in both datasets?
2. What percentage of political news articles (including fake and true news) from the year of 2016 are fake?

***(Hint:*** *To answer this, you will need to search through both datasets and analyse the date and category of each article*.)

1. Which words are most frequently used in fake news articles related to government topics?

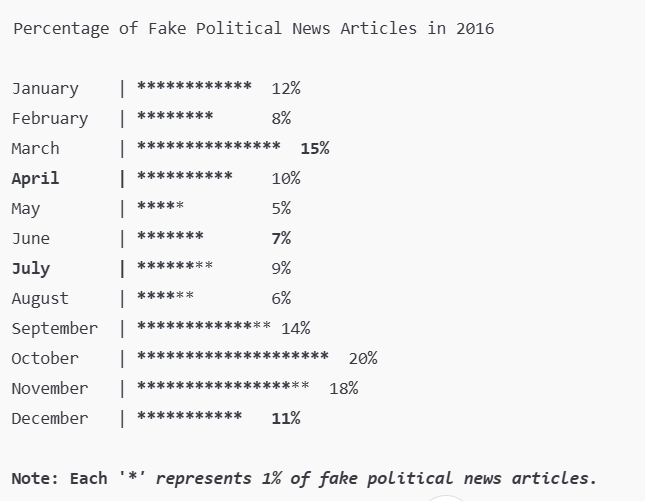
***(Hint:*** *You are required to extract the most common words from these articles, sort them by frequency, and present the results.*)

By completing this assignment, your team will explore the performance of various algorithms and data structures, deepening your team’s understanding of the challenges involved in detecting fake news and how technology can be leveraged to address this growing issue in the digital age.

**Minimum Requirements for Lab Work #1**

1. This assignment may include up to **FOUR (4)** members per team.
2. This assignment requires the development of **two separate programs**:
3. One focusing on **array implementation**
4. Implement an array data structure to store the news articles.
5. Each article will be stored as an object or a structure containing its title, content, category, and publication date.
6. Perform necessary operations such as sorting, searching, and displaying the articles based on certain attributes (e.g., sorting by year).
7. The other on **linked list implementation**
8. Implement a linked list data structure to store the news articles.
9. Each article will be represented by a node, with each node containing the article's title, content, category, and publication date.
10. Similar to the array-based program, ensure that the linked list allows for efficient operations such as sorting, searching, and displaying the articles.
11. Both programs will use data structures to store and organize news articles, including their titles, content, categories, and publication dates, which will be sourced from **TWO (2)** CSV files: *true.csv* and *fake.csv*.
12. **Sort** the articles by their publication year using suitable sorting algorithms for both arrays and linked lists, based on the chosen data structure, **and display** the entire list in ascending order.
13. **Calculate and display** the total number of articles in both datasets (true news and fake news) for each data structure.
14. **Search** the articles based on specific criteria, such as category, publication year, or other factors, using an appropriate searching algorithm for the selected data structure, **and display** the relevant results.
15. Implement a function to analyse and report the percentage of fake political news articles for each month in the year 2016.

Sample output:



1. **Compare** the time and memory efficiency, as well as the performance, of various algorithms and data structures in processing real-world news data from both theoretical and practical perspectives.
2. Your team may also **introduce** new and essential features that utilize both arrays and linked lists within this system, enabling a more **diverse range of comparative analyses** of results, which can provide greater value for the data structures and algorithms aspect.
3. Ensure the inclusion of robust error handling and data validation within the system to effectively address unexpected formats or missing data.

**Submission Guidelines #1: Program and Video Submission – Lab Work #1 (15 Marks)**

1. Your team is required to utilize C++ programming to develop this prototype, comprising two programs in this section.
2. **Built-in containers such as <list>, <vector>, etc. are not allowed** in this assignment. All containers are self-created.

*Refer to the link:* [*https://www.geeksforgeeks.org/containers-cpp-stl/*](https://www.geeksforgeeks.org/containers-cpp-stl/) *for further information on built-in containers in STL C++.*

1. The program will be enhanced if your team effectively implements and compares various useful searching or sorting algorithms in your developed prototype.
2. The evaluation criteria for this lab work #1 include assessing the clarity and structural design of the code, as well as the quality of comments and adherence to good programming practices. (e.g., indentation, meaningful identifier names, comments, etc.).
3. Team leader must submit the system solution in a ZIP file format to Moodle by Friday of Week 9, no later than 5:00 pm.
4. Include only the .cpp, .hpp, and .csv/text files in the ZIP file.
5. The ZIP file must follow this naming format:

*“<GroupNo>\_<Team Leader ID >\_<Member 1 ID >\_<Member 2 ID >\_<Member 3 ID >.zip”*

For example, **“G1\_TP012345\_TP014556\_TP067554\_TP034325.zip”**

1. Each team leader must also upload **ONE (1)**  video recording to the Moodle system by Friday of Week 9, no later than 5:00 pm.
2. The maximum duration for the overall video recording is 30 minutes.
3. All members must participate in the video recording, presenting and explaining the development process in detail, with each member speaking for no more than 5 minutes.
4. Each member must relate their explanation to the workload matrix distribution table provided in the Word document.
5. Compress the final video recording to under 200MB before submitting.
6. If the video exceeds the specified time limit (30 minutes), it will only be assessed up to the specified duration.
7. Videos must be recorded at normal speed (1x) and cannot be sped up or adjusted to meet the demo video duration requirements.
8. The video recording file must adhere to the following name format:

*“<GroupNo>\_<Team Leader ID >\_<Member 1 ID >\_<Member 2 ID >\_<Member 3 ID >.mp4”*

For example, **“G1\_TP012345\_TP014556\_TP067554\_TP034325.mp4”**

1. Refer to the **Page 7** for marking criteria of this Lab Evaluation Work #1 submission.

**Submission Guidelines #2: Documentation Submission – Solution Work (15 Marks)**

1. Create a **Word document** (refer Appendix 1) that explains the overall efficiency and limitations of your developed system
2. Clearly demonstrate the differences in implementation and performance between arrays and linked lists based on the ***operations performed*** in this specific analysis scenario.
3. Present results that clearly show the advantages and disadvantages of using arrays and linked lists in this specific sentiment analysis scenario.
4. Include a summary of all your hypotheses, observations, experimental results, justifications, and reflections, etc. in your word documentation.
5. Your team must include a workload distribution matrix table after the cover page, showing each member's contributions to the project.

*Note that this table will impact each member's personal final mark in Lab Evaluation Work #1 based on their stated contribution percentage.*

1. If you use some code which has been taken or adapted from another source (book, magazine, internet, forum, etc.) then this must be **cited and referenced** using **APA Referencing Style within your source code**, and this must be mentioned explicitly in the **report**. Failure to reference code properly will be treated as plagiarism.
2. The team leader is required to upload a word document before / on Friday of Week 9, no later than 5:00 pm.

The documentation should be named using format:

*“<GroupNo>\_<Team Leader ID >\_<Member 1 ID >\_<Member 2 ID >\_<Member 3 ID >.docx”*

For example, **“G1\_TP012345\_TP014556\_TP067554\_TP034325.docx”**

A screenshot of a computer

Description automatically generated

1. Refer to the **Page 7** for marking criteria of this Solution Work submission.

MARKING CRITERIA

(Lab Evaluation Work #1 - 15 MARKS)

This Lab Evaluation Work #1 will be evaluated according to the following performance criteria:

|  |  |  |
| --- | --- | --- |
| **Assessment Components** | **Inclusive** | **15 Marks** |
| *CLO3: Lab Evaluation Task #1 – 30-Minute Video Recording (****Assessment will be based on individual performance****)* | | |
| Practical Skills: Use of Data Structures & Algorithms + Personal Understanding | | |
| **Utilization of data structures** | Technical Proficiency |  |
| **Implementation of relevant algorithms** | Technical Proficiency |  |
| **Demonstrates understanding of data structures/algorithms used** | Comprehensive Understanding |  |
| **Justifies choices of structures/algorithms** | Insightful Justification |  |

MARKING CRITERIA

(Solution Work - 15 MARKS)

This solution work will be evaluated according to the following performance criteria:

|  |  |  |
| --- | --- | --- |
| **Assessment Components** | **Inclusive** | **15 Marks** |
| *CLO2: Solution Work - Documentation (****Assessment will be based on group component****)* | | |
| **Theoretical Explanation**  (e.g., Data Structures, Algorithms) | Clear explanation |  |
| **Input Output Screenshots** | Adequate Screenshots |  |
| **Summary Discussions**  (Inclusive Time and Space Complexity) | Clear and insightful analysis |  |
| **Conclusion & Reflection**  (Other Relevant / Importance Discussions) | Clearly highlighted and insightful |  |
| **Content Organization** | Well-structured and logical flow |  |

Approximation of Total Pages for the documentation: **30 (max).**

Approximation of Words for the documentation: **4500 words (max)**

**APPENDIX 1:**

**SOLUTION WORK - DOCUMENTATION**

**(15 MARKS)**

The report outline as below:

* **Cover Page** (1 page)
* Module Code and Name
* Intake Code
* Group Number
* Member List.
* **Workload Matrix Table with signature** (1 page)
* **Theoretical Explanation** (2 - 4 pages)
* Elaborate on the data manipulations in chosen data structures.
* **Input Output Screenshots** (5 - 10 pages)
* System input output screenshots
* **Summary Discussions** (5 - 10 pages)
* System efficiency (*including actual execution time, time efficiency, space efficiency etc.*).
* Summarize and discuss all the observations / thoughts made during the development.
* Critically summarize the strengths and weaknesses of the code you developed.
* **Conclusion and Reflection** (1 - 2 pages)
* Summary of your works
* Describe your future work in terms of your system weaknesses, if any.
* Additional personal thoughts on this assignment, if any.
* **References (if applicable)** (1 page)
* **Appendix (if applicable)** (1 page)

Approximation of Total Pages : **30 (max).**

Approximation of Words : **4500 words (max)**