Data Structure and Algorithms(COMP202)
Assignment #2 Programming Assignments

Data Structure and Algorithms(COMP202)

June 2025

Task

1. Write a program to check if any given mathematical expression ha balanced number of parentheses or not?
   Run the program by testing following expression-
   $\Rightarrow a + (b - c) * (d$
   $\Rightarrow m + [a - b * (c + d * \{m)]$
   $\Rightarrow a + (b - c)$
   Hint: Use stack

2. Write a program to convert an infix mathematical expression to postfix and evaluate it.

3. Write and Implement the reverse traversal algorithm in the the linked list. Write main() to demonstrate the use of the function.

4. How can we implement the doubly linked list using structure? Write a method to insert(after given node) and delete the node in doubly linked list. Write main() to demonstrate the use of the functions.

5. How can we implement the undirected graph using adjacency matrix? Write a function that implement the BFS and DFS technique to traverse through the graph. Demonstrate the use of your program with example graph.

6. Given the array of the data (unsorted), Write a program to build the min and max heap.

7. Given a weighted graph, implement Dijkstra's algorithm to find the shortest path from a source node to all other nodes.

8. Write a program that generates a list of N random integers in the range $[1, 1000]$, where N is provided by the user at run time. Then, perform the following tasks:

   (a) Ask the user to choose a sorting algorithm from the following:
   - Bubble Sort
   - Selection Sort
   - Insertion Sort
   - Merge Sort

   (b) Sort the randomly generated numbers using the chosen algorithm.

   (c) Print the numbers before and after sorting.

   (d) Display the total number of comparisons and swaps (if applicable) performed by the chosen algorithm.

Submission Guidelines

1. **Code Structure**
   - Each program must be implemented in a separate source file, named as:

     `prog_<q>.c`

     where `<q>` corresponds to the question number.
   - The code should be properly structured, modular, and well-documented.
2. **Documentation**
   - For each program, a corresponding documentation file must be provided in Markdown format, named as:

     `doc_<q>.md`
   - The documentation must include the following details:
     (a) Explanation of how the data structures are defined.
     (b) Description of the functions implemented, including their purpose.
     (c) An overview of how the `main()` method is organized.
     (d) A sample output of a complete run of the program.
   - **The evaluation will be based primarily on the quality, completeness, and clarity of the documentation.**
3. **Version Control**
   - The complete codebase and documentation must be pushed to a public GitHub repository, with the following repository name:

     `dsa_assignment`
   - Commits should be made gradually throughout the development process, rather than as a single commit submission.
   - Each commit message should be meaningful and clearly describe the changes made.

Academic Integrity

- Plagiarism and use of AI-generated code or text will be strictly checked and penalized.
- Students are expected to submit original work and adhere to the highest standards of academic honesty.