



Course: Data Structures and Algorithms (CC--2042)

Assignment Number 1

Due Date: November 18, 2024.

Semester: Fall-2024

Maximum Marks: 100

Time: 1 Week

Instructor: Dr Bilal Ashfaq Ahmed

Advice to Student

- 1. Read each question carefully and ensure you understand what is required before starting.**
 - 2. Provide well-documented code with comments explaining key sections.**
 - 3. Include test cases where appropriate to demonstrate the functionality of your code.**
 - 4. Write your own code. Do not copy from classmates or online sources.**
-

1. Write a program to compare the number of comparisons made by the binary search and sequential search algorithms as follows:
 - a. Use a random number generator to fill an array list of 1000 elements.
 - b. Use any sorting algorithm to sort list.
 - c. Search list for some items using:
 - i. Binary search algorithm. Modify the algorithm to count the number of comparisons.
 - ii. Binary search algorithm that switches to sequential search when the size of the search list reduces to less than 15. Use a sequential search algorithm suitable for a sorted list.
 - d. Print the number of comparisons for steps c(i) and c(ii). If the item is found in the list, then print its position.
2. Write a program by including a function that rotates an array of integers to the left by a given number k. For example, given the array [1, 2, 3, 4, 5] and k = 2, the array becomes [3, 4, 5, 1, 2].
3. Write a program that identifies all duplicate elements in an integer array and counts how many times each duplicate element occurs.
4. Write a program by including the function insert of the class orderedLinkedList does not check if the item to be inserted is already in the list; that is, it does not check for duplicates. Rewrite the definition of the function insert so that before inserting the item, it checks whether the item to be inserted is already in the list. If the item to be inserted is already in the list, the function outputs an appropriate error message.

5. Write a program to Swap two adjacent elements by adjusting only the links (and not the data) using
 - a. singly linked lists
 - b. doubly linked lists
6. Write a simple airline ticket reservation program. The program should display a menu with the following options: reserve a ticket, cancel a reservation, check whether a ticket is reserved for a particular person, and display the passengers. The information is maintained on an alphabetized linked list of names. In a simpler version of the program, assume that tickets are reserved for only one flight. In a fuller version, place no limit on the number of flights. Create a linked list of flights with each node including a pointer to a linked list of passengers.
7. The Josephus problem is the following game: N people, numbered 1 to N , are sitting in a circle. Starting at person 1, a hot potato is passed. After M passes, the person holding the hot potato is eliminated, the circle closes ranks, and the game continues with the person who was sitting after the eliminated person picking up the hot potato. The last remaining person wins. Thus, if $M = 0$ and $N = 5$, players are eliminated in order, and player 5 wins. If $M = 1$ and $N = 5$, the order of elimination is 2, 4, 1, 5.
 - a. Write a program to solve the Josephus problem for general values of M and N . Try to make your program as efficient as possible. Make sure you dispose of cells.
 - b. What is the running time of your program?
 - c. If $M = 1$, what is the running time of your program? How is the actual speed affected by the delete routine for large values of N ($N > 100,000$)?
8. Write a program by including a function that merges two sorted singly linked lists into a single sorted linked list without creating new nodes (i.e., by rearranging the links). Test your function with different lists.
9. Write a program by including a function to find the middle node of a singly linked list without using the length of the list. If the list has an even number of nodes, return the first of the two middle nodes.

10. Implement a doubly circular linked list class in C++. Include functions to:

- a. Insert a node at the beginning, end, and at a specific position.
- b. Delete a node from the beginning, end, and a specific position.
- c. Traverse the list in both forward and backward directions starting from any node.
- d. Search for a value in the list.

Write a program to test all the functions of your class. Ensure that the circular nature of the list is maintained after each operation.

BEST OF LUCK