Tutorial 2

1) Write an algorithm to
    a. Delete duplicate elements in an unsorted array. What will be the best and worst-case time complexity?
    b. Delete duplicate elements in a sorted array. What will be the best and worst-case time complexity?
2) Write an algorithm to merge two sorted arrays and find the median element.
3) Given a sorted array containing $\boldsymbol{n}$ numbers, write an $\boldsymbol{\Theta(n)}$ algorithm to determines whether or not there exist two elements in the array whose sum is exactly $\boldsymbol{x}$.
    - **Example:** Given array [1, 4, 45, 6, 10, -8] and x = 16
    - **Output:** Yes (6 and 10)
4) Write an algorithm to print the key value of n^{th} node from the end of a linked list. The complexity of the algorithm should be O(n).
5) Write an algorithm to reverse the nodes of a linked list in pairs. If you are given a linked list that holds 1→ 2 → 3 → 4, then after reversal the linked list should hold 2→ 1 → 4 → 3.
6) The time complexity of a simple Bubble Sort is $O(n^2)$ even when the array is already sorted. Write an improved Bubble Sort algorithm which has a time complexity of $O(n)$ in the best case (already sorted).
7) Write an $O(n^2)$ algorithm for finding the element which appears the maximum number of times in an array.
8) Given two polynomials as follows

$$y_1 = 5x^4 + 3x^3 + 2x$$

$$y_2 = 4x^3 + 6x + 7$$

   How you can represent these polynomials using linked lists? Write an $O(n)$ algorithm to add these polynomials.

   Example: $y_1 + y_2 = 5x^4 + 7x^3 + 8x + 7$

9) Given a sorted array of n integers that has been rotated an unknown number of times, write an $O(log\ n)$ algorithm that finds a particular element in the array.
    - **Example:** Find 5 in array [15, 16, 19, 20, 25, 1, 3, 4, 5, 7, 10, 14]
    - **Output:** Index 9

Optional Programming Assignments

→Write programs instead of algorithms for all the above problems (1-9). You can use any programming language.