Week 2

1. Read the degree of two polynomials and their coefficients, all integers, from the standard input. The polynomial is of the form ( ) = ∗ + ⋯ + 1 ∗ 1 +, where 0 ≠ 0.
   1. Write the pseudo code for adding two polynomials.

Step1: Start

Step2: int[] add(int num1[], int num2[], int p, int q) {

        int size = max(p, q);

        int sum[] = new int[size];

        for (int i = 0; i < p; i++) {

            sum[i] = num1[i];

        }

        for (int i = 0; i < q; i++) {

            sum[i] += num2[i];

        }

        return sum;

    }

Step3: printPoly(int poly[ ], int q) {

        for (int i = 0; i < q; i++) {

            System.out.print(poly[i]);

            if (i != 0) {

                System.out.print("x^" + i);

            }

            if (i != q - 1) {

                System.out.print(" + ");

            }

        }

    }

Step4: int num1[] = {5, 0, 10, 6};

        int num2[] = {1, 2, 4};

        int p = num1.length;

        int q = num2.length;

        System.out.println("First polynomial is");

        printPoly(num1, p);

        System.out.println("\nSecond polynomial is");

        printPoly(num2, q);

        int sum[] = add(num1, num2, p, q);

        int size = max(p, q);

        System.out.println("\nsum polynomial is");

        printPoly(sum, size);

    }

}

step5: Stop

1. Write the pseudo code and code for a function that determines whether given word is palindrome. What is the time complexity (expressed using Big O notation)?

Pseudo code of palindrome

Step 1: Start

Step 2: Declare variable string, reverse

Step 3: Read values string, reverse input from user

Step 4: for ( int i =length-1; I >=0; i--)

reverse = reverse + string.charAttri(i);

if(str.equals(reverse)

System.Out.Println(string+"is a palindrome");

else

System.Out.Println(string+"is not a palindrome");

Step 5: Repeat Step 4

Step 6: Display Palindrome

Step 7: Stop

Code for Palindrome Function

public static void main(String[] args);

{

String string, reverse="";

string = sc.nextline();

int length = string,length();

for(int i = length - 1; i>=0; i--)

reverse = reverse + string.charAttri(i);

if(string.equal(reverse))

System.Out.Println(string+"is a palindrome");

else

System.Out.Println(string+"is not a palindrome");

}

Time complexity of Big Notation is an amount of time taken by an algorithm to run program. Some common time complexities are:

1. O(1) - Constant time complexity

2. O(n) - Linear time complexity

3. O(log n) - Logarithmic time complexity

4. O(n^2) - Quadratic time complexity