Maha Gowri S **14/11/2024**

1.Stock Buy and sell

Code:

```
import java.util.ArrayList;
import java.util.Scanner;

class Interval {
    int buy;
    int sell;
}

class Stock {
    ArrayList<ArrayList<Integer>> stockBuySell(int A[], int n) {
        ArrayList<ArrayList<Integer>> result = new ArrayList<>();

        if (n == 1) {
            return result;
        }

        ArrayList<Interval> stock = new ArrayList<>();
        int i = 0, cnt = 0;

        while (i < n - 1) {
            while ((i < n - 1) && (A[i + 1] <= A[i])) {
                i++;
            }

            if (i == n - 1) {
                break;
```

```
        }

        Interval e = new Interval();
        e.buy = i++;

        while ((i < n) && (A[i] >= A[i - 1])) {
            i++;
        }

        e.sell = i - 1;
        stock.add(e);
        cnt++;
    }

    if (cnt == 0) {
        return result;
    } else {
        for (int j = 0; j < stock.size(); j++) {
            result.add(new ArrayList<>());
            result.get(j).add(0, stock.get(j).buy);
            result.get(j).add(1, stock.get(j).sell);
        }
    }
    return result;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the number of days: ");
        int n = scanner.nextInt();

        int[] prices = new int[n];
        System.out.println("Enter the stock prices for each day: ");
        for (int i = 0; i < n; i++) {
            prices[i] = scanner.nextInt();
        }

        Stock stock = new Stock();
        ArrayList<ArrayList<Integer>> result = stock.stockBuySell(prices, n);

        if (result.isEmpty()) {
            System.out.println("No profit can be made.");
        } else {
            System.out.println("Buy and Sell Days:");
            for (ArrayList<Integer> interval : result) {
                System.out.println("Buy on day: " + interval.get(0) + ", Sell on day: " +
interval.get(1));
            }
        }

        scanner.close();
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Stock
Enter the number of days: 7
Enter the stock prices for each day:
100 180 260 310 40 535 695
Buy and Sell Days:
Buy on day: 0, Sell on day: 3
Buy on day: 4, Sell on day: 6

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>
```

Time Complexity:O(n)

2.Coin Change (Count Ways)

Code:

```
import java.util.*;


class Coins {
    static int count(int[] coins, int sum, int n, int[][] dp) {
        if (sum == 0)
            return dp[n][sum] = 1;
        if (n == 0 || sum < 0)
            return 0;
        if (dp[n][sum] != -1)
            return dp[n][sum];
        return dp[n][sum] = count(coins, sum - coins[n - 1], n, dp) + count(coins, sum, n
- 1, dp);
    }


    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter the number of coins: ");
        int n = scanner.nextInt();

        int[] coins = new int[n];
        System.out.print("Enter the values of the coins: ");
        for (int i = 0; i < n; i++) {
            coins[i] = scanner.nextInt();
        }

        System.out.print("Enter the sum to achieve: ");
        int sum = scanner.nextInt();

        int[][] dp = new int[n + 1][sum + 1];
        for (int[] row : dp)
            Arrays.fill(row, -1);

        int res = count(coins, sum, n, dp);
        System.out.println("Number of ways to make the sum: " + res);
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>javac Coins.java

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Coins
Enter the number of coins: 3
Enter the values of the coins: 1 2 3
Enter the sum to achieve: 4
Number of ways to make the sum: 4

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Coins
Enter the number of coins: 4
Enter the values of the coins: 2 5 3 6
Enter the sum to achieve: 10
Number of ways to make the sum: 5

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>
```

Time Complexity: O(N*Sum)

3.First and Last Occurrence

Code:

```
mport java.util.Scanner;


class Fal {
    public static void findFirstAndLast(int arr[], int x) {
        int n = arr.length;
        int first = -1, last = -1;
        for (int i = 0; i < n; i++) {
            if (x != arr[i])
                continue;
            if (first == -1)
                first = i;
            last = i;
        }
        if (first != -1) {
            System.out.println("First Occurrence = " + first);
```

```
            System.out.println("Last Occurrence = " + last);
        } else {
            System.out.println("Not Found");
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();

        int[] arr = new int[n];
        System.out.print("Enter the elements of the array (sorted): ");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        System.out.print("Enter the element to find: ");
        int x = scanner.nextInt();

        findFirstAndLast(arr, x);
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>javac FAL.java

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java FAL
Error: Could not find or load main class FAL
 Fal)

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Fal
Enter the number of elements in the array: 9
Enter the elements of the array (sorted): 1 3 5 5 5 5 67 123 125
Enter the element to find: 5
First Occurrence = 2
Last Occurrence = 5

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>
```

Time Complexity:O(N)

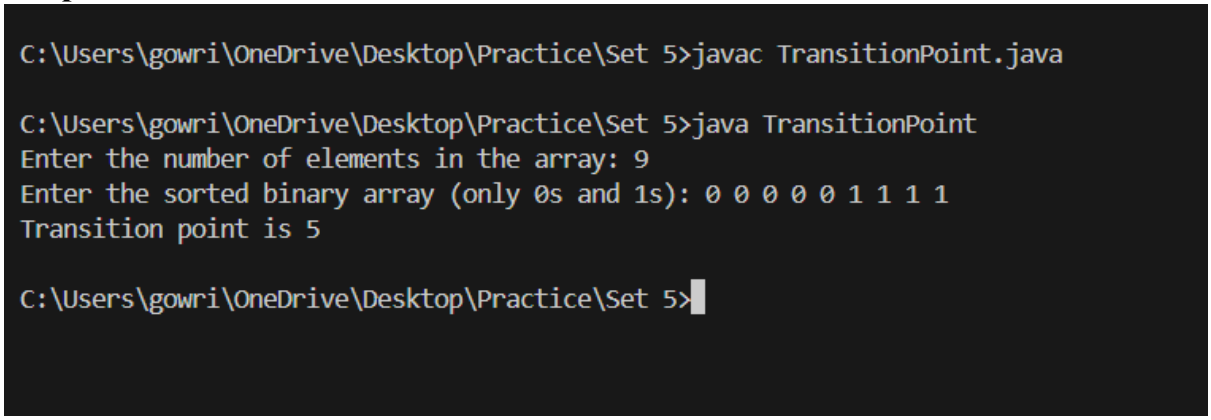
4.Transition Point

Code:

```
import java.util.Scanner;


class TransitionPoint {
    static int findTransitionPoint(int arr[], int n) {
        int lb = 0, ub = n - 1;


        while (lb <= ub) {
            int mid = (lb + ub) / 2;


            if (arr[mid] == 0)
                lb = mid + 1;
            else if (arr[mid] == 1) {
                if (mid == 0 || (mid > 0 && arr[mid - 1] == 0))
                    return mid;
                ub = mid - 1;
```

```
            }
        }
        return -1;
    }

    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of elements in the array: ");
        int n = scanner.nextInt();

        int[] arr = new int[n];
        System.out.print("Enter the sorted binary array (only 0s and 1s): ");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        int point = findTransitionPoint(arr, n);
        System.out.println(point >= 0 ? "Transition point is " + point : "There is no transition point");
    }
}
```

Output:



Time Complexity: O(log N)

5.Find Repeating Elements

Code:

```
import java.util.*;

class RepeatingElements {
    static void printFirstRepeating(int arr[]) {
        int min = -1;
        HashSet<Integer> set = new HashSet<>();

        for (int i = arr.length - 1; i >= 0; i--) {
            if (set.contains(arr[i]))
                min = i;
            else
                set.add(arr[i]);
        }

        if (min != -1)
            System.out.println("The first repeating element is " + arr[min]);
        else
            System.out.println("There are no repeating elements");
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of elements in the array:");
        int n = scanner.nextInt();
        int[] arr = new int[n];
```

```
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        printFirstRepeating(arr);
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java RepeatingElements
Enter the number of elements in the array:
7
Enter the elements of the array:
The first repeating element is 5

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java RepeatingElements
Enter the number of elements in the array:
9
Enter the elements of the array:
6 10 5 4 9 120 4 6 10
The first repeating element is 6

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>
```

Time Complexity:O(n)

6.Remove Duplicates Sorted Array

Code:

```
import java.util.HashSet;
import java.util.Scanner;

class Duplicates {

    static int removeDuplicates(int[] arr) {
```

```
    HashSet<Integer> s = new HashSet<>();
    int idx = 0;

    for (int i = 0; i < arr.length; i++) {
        if (!s.contains(arr[i])) {
            s.add(arr[i]);
            arr[idx++] = arr[i];
        }
    }

    return idx;
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter the number of elements in the array:");
    int n = scanner.nextInt();
    int[] arr = new int[n];

    System.out.println("Enter the elements of the array:");
    for (int i = 0; i < n; i++) {
        arr[i] = scanner.nextInt();
    }

    int newSize = removeDuplicates(arr);

    System.out.println("Array after removing duplicates:");
    for (int i = 0; i < newSize; i++) {
        System.out.print(arr[i] + " ");
```

```
        }
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>javac Duplicates.java

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Duplicates
Enter the number of elements in the array:
7
Enter the elements of the array:
1 2 3 4 3 4 5
Array after removing duplicates:
1 2 3 4 5
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Duplicates
Enter the number of elements in the array:
9
Enter the elements of the array:
4 5 67 3 4 8 4 67 67
Array after removing duplicates:
4 5 67 3 8
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>
```

Time Complexity: O(n)

7.Maximum Index

Code:

```
import java.util.Scanner;


public class Maximum {


    int maxIndexDiff(int arr[], int n) {
        int maxDiff = -1;
        for (int i = 0; i < n; ++i) {
            for (int j = n - 1; j > i; --j) {
                if (arr[j] > arr[i] && maxDiff < (j - i))
                    maxDiff = j - i;
            }
```

```
        }
        return maxDiff;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the number of elements in the array:");
        int n = scanner.nextInt();
        int[] arr = new int[n];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            arr[i] = scanner.nextInt();
        }

        Maximum max = new Maximum();
        int maxDiff = max.maxIndexDiff(arr, n);
        System.out.println("Maximum difference is: " + maxDiff);
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>javac Maximum.java

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Maximum.java
Enter the number of elements in the array:
9
Enter the elements of the array:
34 8 10 3 2 80 30 33 1
Maximum difference is: 6

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java Maximum
```

Time Complexity:O(N)

8.Wave Array

Code:

```
import java.util.*;

class WaveSort
{
   void swap(int arr[], int a, int b)
   {
      int temp = arr[a];
      arr[a] = arr[b];
      arr[b] = temp;
   }

   void sortInWave(int arr[], int n)
   {
      Arrays.sort(arr);

      for (int i=0; i<n-1; i += 2)
         swap(arr, i, i+1);
   }

   public static void main(String args[])
   {
      WaveSort ob = new WaveSort();
      int arr[] = {10, 90, 49, 2, 1, 5, 23};
      int n = arr.length;
      ob.sortInWave(arr, n);
      for (int i : arr)
         System.out.print(i + " ");
```

```
    }
}
```

Output:

```
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java WaveSort
Enter the number of elements in the array:
7
Enter the elements of the array:
10 90 49 2 1 5 23
Array in wave form:
90 10 49 1 5 2 23
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>javac WaveSort.java

C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>java WaveSort
2 1 10 5 49 23 90
C:\Users\gowri\OneDrive\Desktop\Practice\Set 5>
```

Time Complexity:O(N)