Name: Tejaswini Anil Kamble

Email: teju000kamble@gmail.com

DAY 12: Assignments

Task 1: Bit Manipulation Basics

Create a function that counts the number of set bits (1s) in the binary representation of an integer. Extend this to count the total number of set bits in all integers from 1 to n.

Ans: Source code

```
FS JavaProgramming - Data strutures/src/com/wipro/day12/BitManipulation.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Q 🔡 🕸 🐉
🔓 Project Explorer 🗶 📅 🗖 🗓 Krushkal Algorithm.ja... 📗 Kruskals Algorithm.ja... 📗 MyNew Graph.java 📗 MyEdge.java 📗 MyCycle Detection.java 📗 MyUnion Find.java
  ☐ Use System Library [JavaSE-3] public class BitManipulation {
                               public static int countSetBits(int x) {
                                    int count = 0;
    > 🔠 com.ds.graph
    > # com.ds.linkedlist
> # com.ds.patterns
                                         count += x & 1;
    > 🌐 com.ds.queue
    > # com.ds.searching_algori 9
> # com.ds.sortingAlgorithr 10
                                       return count;
    > # com.ds.stack
    > # com.us.stack
> # com.ds.timeAndSpaceC 11
12

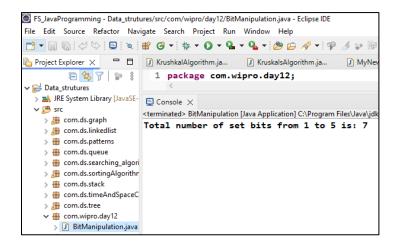
                                 public static int totalSetBits(int n) {
    > # com.ds.tree
    > the com.ds.tree

- com.wipro.day12

- BitManipulation.java

- com.wipro.graphalgo
                                       int totalCount = 0;
                                       for (int i = 1; i <= n; i++) {
                                            totalCount += countSetBits(i);
    > 🔠 com.wipro.graphalgo
                        16
 > 📂 DSA_JavaAssignments
                          18
                                  public static void main(String[] args) {
 Firstjava 😅
                                       System.out.println("Total number of set bits from 1 to " + n + " is: " + totalSetBits(n)); // Output: 7
                          23 }
```

Output:



Task 2: Unique Elements Identification

Given an array of integers where every element appears twice except for two, write a function that efficiently finds these two non-repeating elements using bitwise XOR operations.

Ans: Source code

```
FS_JavaProgramming - Data_strutures/src/com/wipro/day12/UniqueElements.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Q 🔡 😭 🐉
🌇 Project Explorer 🗴 🕝 🗖 📝 Krushkal Algor... 📝 Kruskals Algor... 🧷 MyNewGraph.java 📝 MyEdgejava 📝 MyCycle Detect... 📝 MyUnion Find.j... 📝 BitManipulati... 📝 Unique Element... 🗴 🖳
          package com.wipro.day12;
  Some surfutures

> max FE System Library [JavaSE-3 public class UniqueElements {

> max FE System Library [JavaSE-3 public class UniqueElements {
                           4
5⊖
6
7
    > 🔠 com.ds.graph
> 🔠 com.ds.linkedlist
                                      public static int[] findUniqueElements(int[] nums) {
                                         int xor_all = 0;
for (int num : nums) {
    xor_all ^= num;
     > # com.ds.patterns
                                        }
     > # com.ds.sortingAlgorithr 10
    > # com.ds.stack 11
> # com.ds.timeAndSpaceC 12
                                        int setBit = xor_all & -xor_all;
      # com.ds.tree
    > acom.ds.tree

v com.wipro.day12

) BitManipulation.java

) UniqueElements.java

) Com.wipro.graphalgo
                                           int unique1 = 0, unique2 = 0;
                                           for (int num : nums) {
   if ((num & setBit) == 0) {
                                               unique1 ^= num;
} else {
     > # com.wipro.graphalgo
> / module-info.java
                             17
                                                     unique2 ^= num;
 ≥ DSA_JavaAssignments
                             19
  ems
                            20
21
                                          }
  mvcstruture
                             22
                                           return new int[]{unique1, unique2};
                             23
                             24
                                      public static void main(String[] args) {
                                           int[] nums = {2, 4, 7, 9, 2, 4, 5, 7};
int[] result = findUniqueElements(nums);
                             26
                                           System.out.println("The two unique numbers are: " + result[0] + " and " + result[1]);
                             28
                             29
                             30 }
                            31
                                                            🚃 💼 客 🧿 🧽 🞹 🕞 👑 🞹 🕼 🐉 📆 🔯 🛗 🙀 🔯 🙀 🙀 15°C 🛆 🖟 🖟 🐧 40) 🖦 🖟 ENG 1-01 PM
       Type here to search
```

Output:

