CSE3004 Design and Analysis of Algorithms ELA Winter 2023-2024 Semester

Lab sheet - (L15+L16)

Name : Koya Madhusudhana Rao

Reg.No : 21BCE9905

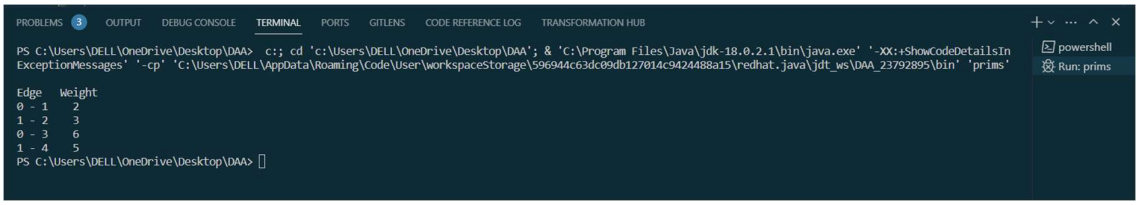
Git Hub Repository :- https://github.com/Koya-Madhusudhana-Rao/DAA-Lab

1. Write a programs to implement the following:

a) Prim's algorithm.

```java
import java.util.Arrays;

public class prims {

    // Function to find the minimum spanning tree using Prim's algorithm
    public static void primMST(int[][] graph) {
        int vertices = graph.length;

        // Arra y to store the parent of each vertex in the MST
        int[] parent = new int[vertices];

        // Array to store the key values of each vertex
        int[] key = new int[vertices];

        // Array to keep track of whether a vertex is included in the MST
        boolean[] mstSet = new boolean[vertices];

        // Initialize key values to infinity and mstSet to false
        Arrays.fill(key, Integer.MAX_VALUE);
        Arrays.fill(mstSet, false);

        // Start with the first vertex
        key[0] = 0;
        parent[0] = -1;

        // Construct the MST
        for (int count = 0; count < vertices - 1; count++) {
            int u = minKey(key, mstSet);
            mstSet[u] = true;
```

```
            // Update key values and parent for adjacent vertices of the
chosen vertex
            for (int v = 0; v < vertices; v++) {
                if (graph[u][v] != 0 && !mstSet[v] && graph[u][v] < key[v]) {
                    parent[v] = u;
                    key[v] = graph[u][v];
                }
            }
        }

        // Print the MST
        System.out.println("Edge   Weight");
        for (int i = 1; i < vertices; i++) {
            System.out.println(parent[i] + " - " + i + "    " +
graph[i][parent[i]]);
        }
    }

    // Function to find the vertex with the minimum key value
    private static int minKey(int[] key, boolean[] mstSet) {
        int min = Integer.MAX_VALUE, minIndex = -1;
        int vertices = key.length;

        for (int v = 0; v < vertices; v++) {
            if (!mstSet[v] && key[v] < min) {
                min = key[v];
                minIndex = v;
            }
        }

        return minIndex;
    }

    // Example usage
    public static void main(String[] args) {
        int[][] graph = {
                {0, 2, 0, 6, 0},
                {2, 0, 3, 8, 5},
                {0, 3, 0, 0, 7},
                {6, 8, 0, 0, 9},
                {0, 5, 7, 9, 0}
        };

        primMST(graph);
    }
}
```



b) Kruskal's algorithm.

```java
import java.util.Arrays;
import java.util.Comparator;

class Edge {
    int src, dest, weight;

    Edge(int src, int dest, int weight) {
        this.src = src;
        this.dest = dest;
        this.weight = weight;
    }
}

class KruskalAlgorithm {

    // Function to find the minimum spanning tree using Kruskal's algorithm
    public static void kruskalMST(int vertices, Edge[] edges) {
        Arrays.sort(edges, Comparator.comparingInt(o -> o.weight));

        int[] parent = new int[vertices];
        for (int i = 0; i < vertices; i++) {
            parent[i] = i;
        }

        System.out.println("Edge   Weight");
        for (Edge edge : edges) {
            int rootSrc = find(parent, edge.src);
            int rootDest = find(parent, edge.dest);

            if (rootSrc != rootDest) {
                System.out.println(edge.src + " - " + edge.dest + "    " +
edge.weight);
                union(parent, rootSrc, rootDest);
            }
        }
    }

    // Function to find the root of a set using path compression
    private static int find(int[] parent, int vertex) {
        if (parent[vertex] != vertex) {
```

```
            parent[vertex] = find(parent, parent[vertex]);
        }
        return parent[vertex];
    }

    // Function to perform union of two sets
    private static void union(int[] parent, int x, int y) {
        int rootX = find(parent, x);
        int rootY = find(parent, y);
        parent[rootX] = rootY;
    }

    // Example usage
    public static void main(String[] args) {
        int vertices = 5;
        Edge[] edges = {
                new Edge(0, 1, 2),
                new Edge(0, 3, 6),
                new Edge(1, 2, 3),
                new Edge(1, 3, 8),
                new Edge(1, 4, 5),
                new Edge(2, 4, 7),
                new Edge(3, 4, 9)
        };

        kruskalMST(vertices, edges);
    }
}
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  CODE REFERENCE LOG  TRANSFORMATION HUB

PS C:\Users\DELL\OneDrive\Desktop\DAA>  c:; cd 'c:\Users\DELL\OneDrive\Desktop\DAA'; & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.e
xe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\596944c63dc09db127014c9
424488a15\redhat.java\jdt_ws\DAA_23792895\bin' 'KruskalAlgorithm'
Edge   Weight
0 - 1    2
1 - 2    3
1 - 4    5
0 - 3    6
PS C:\Users\DELL\OneDrive\Desktop\DAA>

powershell
Run: prims
Run: Kruskal...
```

2. Create a Java class called Student with the following details as variables within it. (i) USN (ii)

Name (iii) Programme (iv) Phone. Write a Java program to create n Student objects and print the

USN, Name, Programme, and Phone of these objects with suitable headings.

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

```java
class Student {
    String usn;
    String name;
    String programme;
    String phone;

    // Constructor
    public Student(String usn, String name, String programme, String phone) {
        this.usn = usn;
        this.name = name;
        this.programme = programme;
        this.phone = phone;
    }

    // Method to display student information
    public void displayInfo() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Programme: " + programme);
        System.out.println("Phone: " + phone);
        System.out.println();
    }
}

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

        System.out.print("Enter the number of students (n): ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        // Create an array to hold n Student objects
        Student[] students = new Student[n];

        // Input information for each student
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Student " + (i + 1) +
":");
            System.out.print("Enter USN: ");
            String usn = scanner.nextLine();
            System.out.print("Enter Name: ");
            String name = scanner.nextLine();
            System.out.print("Enter Programme: ");
            String programme = scanner.nextLine();
            System.out.print("Enter Phone: ");
            String phone = scanner.nextLine();
```

```
            // Create a new Student object with the entered details
            students[i] = new Student(usn, name, programme, phone);
        }

        // Display information for each student
        System.out.println("\nStudent Information:");
        for (Student student : students) {
            student.displayInfo();
        }
    }
}
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  CODE REFERENCE LOG  TRANSFORMATION HUB

 '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\596944c63dc09db127014c9424488a15\redhat.java\jdt_ws\DAA_23792895\bin
' 'StudentTest'
Enter the number of students (n): 3

Enter details for Student 1:
Enter USN: 12345
Enter Name: Madhu
Enter Programme: CSE
Enter Phone: 123456

Enter details for Student 2:
Enter USN: 12345
Enter Name: Vinay
Enter Programme: CSE
Enter Phone: 1234567

Enter details for Student 3:
Enter USN: 12345
Enter Name: Bhaskar
Enter Programme: CSE
Enter Phone: 12345678

Student Information:
USN: 12345
Name: Madhu
Programme: CSE
Phone: 123456

USN: 12345
Name: Vinay
Programme: CSE
Phone: 1234567

USN: 12345
Name: Bhaskar
Programme: CSE
Phone: 12345678

PS C:\Users\DELL\OneDrive\Desktop\DAA>
```

3. Design a superclass called Staff with details as StaffId, Name, Phone, Salary. Extend this class

by writing three subclasses namely Teaching (domain, publications), Technical (skills), and

Contract (period). Write a Java program to read and display at least 3 staff objects of all three

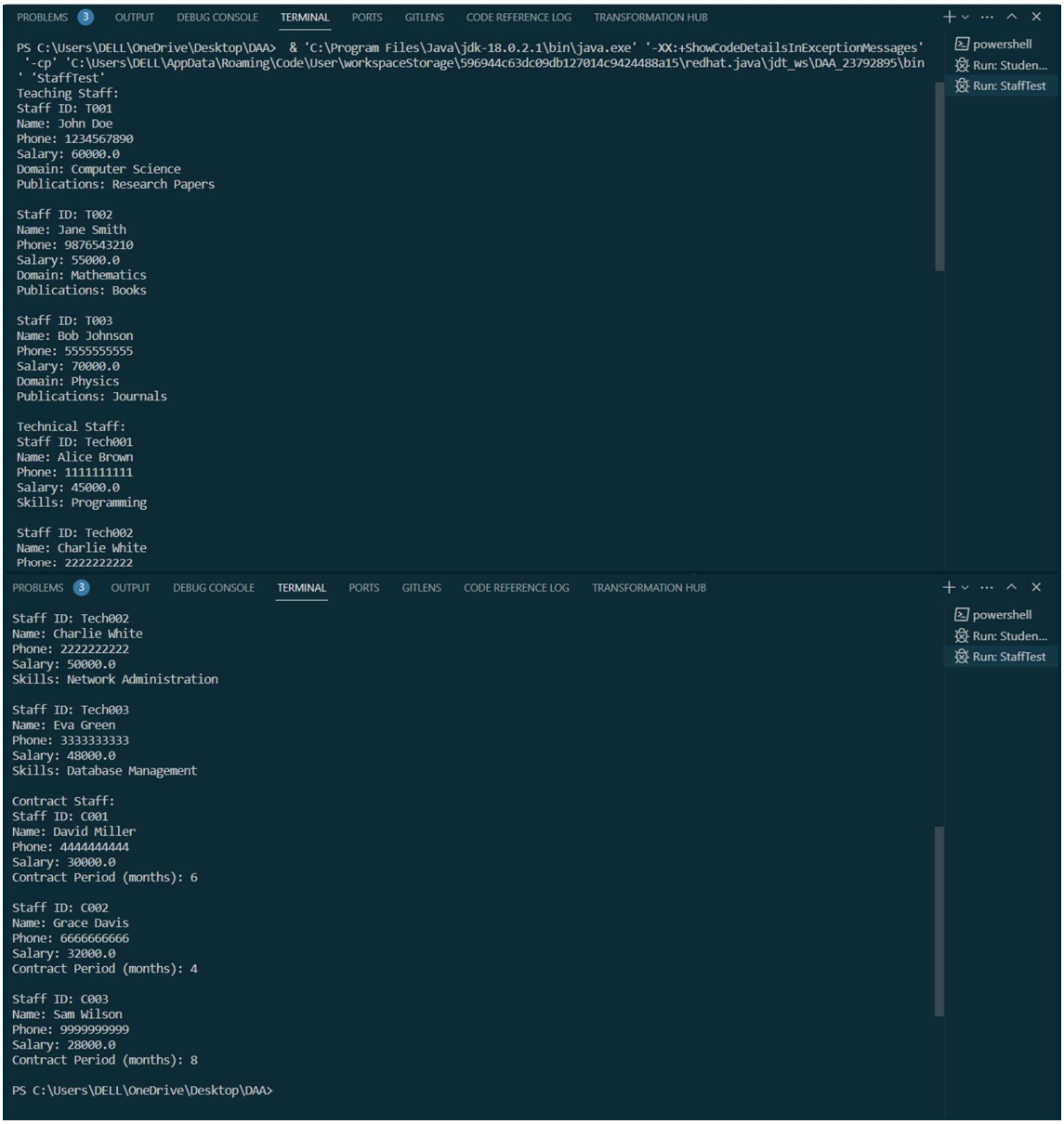
categories

```
import java.util.Scanner;

class Staff {
    String staffId;
    String name;
```

```
    String phone;
    double salary;

    // Constructor
    public Staff(String staffId, String name, String phone, double salary) {
        this.staffId = staffId;
        this.name = name;
        this.phone = phone;
        this.salary = salary;
    }

    // Method to display staff information
    public void displayInfo() {
        System.out.println("Staff ID: " + staffId);
        System.out.println("Name: " + name);
        System.out.println("Phone: " + phone);
        System.out.println("Salary: " + salary);
    }
}

class Teaching extends Staff {
    String domain;
    String publications;

    // Constructor
    public Teaching(String staffId, String name, String phone, double salary,
String domain, String publications) {
        super(staffId, name, phone, salary);
        this.domain = domain;
        this.publications = publications;
    }

    // Method to display teaching staff information
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Domain: " + domain);
        System.out.println("Publications: " + publications);
        System.out.println();
    }
}

class Technical extends Staff {
    String skills;

    // Constructor
    public Technical(String staffId, String name, String phone, double salary,
String skills) {
```

```
        super(staffId, name, phone, salary);
        this.skills = skills;
    }

    // Method to display technical staff information
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Skills: " + skills);
        System.out.println();
    }
}

class Contract extends Staff {
    int period;

    // Constructor
    public Contract(String staffId, String name, String phone, double salary,
int period) {
        super(staffId, name, phone, salary);
        this.period = period;
    }

    // Method to display contract staff information
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Contract Period (months): " + period);
        System.out.println();
    }
}

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

        // Create at least 3 staff objects for each category
        Teaching teachingStaff1 = new Teaching("T001", "John Doe",
"1234567890", 60000.0, "Computer Science", "Research Papers");
        Teaching teachingStaff2 = new Teaching("T002", "Jane Smith",
"9876543210", 55000.0, "Mathematics", "Books");
        Teaching teachingStaff3 = new Teaching("T003", "Bob Johnson",
"5555555555", 70000.0, "Physics", "Journals");

        Technical technicalStaff1 = new Technical("Tech001", "Alice Brown",
"1111111111", 45000.0, "Programming");
        Technical technicalStaff2 = new Technical("Tech002", "Charlie White",
"2222222222", 50000.0, "Network Administration");
```

```
        Technical technicalStaff3 = new Technical("Tech003", "Eva Green",
"3333333333", 48000.0, "Database Management");

        Contract contractStaff1 = new Contract("C001", "David Miller",
"4444444444", 30000.0, 6);
        Contract contractStaff2 = new Contract("C002", "Grace Davis",
"6666666666", 32000.0, 4);
        Contract contractStaff3 = new Contract("C003", "Sam Wilson",
"9999999999", 28000.0, 8);

        // Display information for each staff object
        System.out.println("Teaching Staff:");
        teachingStaff1.displayInfo();
        teachingStaff2.displayInfo();
        teachingStaff3.displayInfo();

        System.out.println("Technical Staff:");
        technicalStaff1.displayInfo();
        technicalStaff2.displayInfo();
        technicalStaff3.displayInfo();

        System.out.println("Contract Staff:");
        contractStaff1.displayInfo();
        contractStaff2.displayInfo();
        contractStaff3.displayInfo();
    }
}
```


PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS CODE REFERENCE LOG TRANSFORMATION HUB
PS C:\Users\DELL\OneDrive\Desktop\DAA> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\DELL\AppData\Roaming\Code\User\workspaceStorage\596944c63dc09db127014c9424488a15\redhat.java\jdt_ws\DAA_23792895\bin' 'StaffTest'
Teaching Staff:
Staff ID: T001
Name: John Doe
Phone: 1234567890
Salary: 60000.0
Domain: Computer Science
Publications: Research Papers
Staff ID: T002
Name: Jane Smith
Phone: 9876543210
Salary: 55000.0
Domain: Mathematics
Publications: Books
Staff ID: T003
Name: Bob Johnson
Phone: 5555555555
Salary: 70000.0
Domain: Physics
Publications: Journals
Technical Staff:
Staff ID: Tech001
Name: Alice Brown
Phone: 1111111111
Salary: 45000.0
Skills: Programming
Staff ID: Tech002
Name: Charlie White
Phone: 2222222222
powershell
Run: Studen...
Run: StaffTest
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS CODE REFERENCE LOG TRANSFORMATION HUB
Staff ID: Tech002
Name: Charlie White
Phone: 2222222222
Salary: 50000.0
Skills: Network Administration
Staff ID: Tech003
Name: Eva Green
Phone: 3333333333
Salary: 48000.0
Skills: Database Management
Contract Staff:
Staff ID: C001
Name: David Miller
Phone: 4444444444
Salary: 30000.0
Contract Period (months): 6
Staff ID: C002
Name: Grace Davis
Phone: 6666666666
Salary: 32000.0
Contract Period (months): 4
Staff ID: C003
Name: Sam Wilson
Phone: 9999999999
Salary: 28000.0
Contract Period (months): 8
PS C:\Users\DELL\OneDrive\Desktop\DAA>
powershell
Run: Studen...
Run: StaffTest