

# LAB NETWORK CABLING

PDSA HW9

**Release date:** 05/20 16:00

**TA hour:** 05/27 13:30

**Due date:** 05/29 21:00

Search ...



## HW9 - Lab Network Cabling

Your lab advisor aims to create an ideal study environment with stable network connections for the router, the server, printers, and students' computers. To reduce cost and ensure optimal communication, **the network design should minimize the total cable length while connecting all devices**. Given your background in 'Practical Data Structures and Algorithms,' your advisor has assigned you the task of designing the lab's cabling.



## Description

After implementing Minimum Spanning Tree for network planning based on device information and the distances between devices, your advisor asks you:

- **The Total Cabling Cost :** This refers to the overall cost of cabling, with the assumption that each unit length of cable costs 1 dollar.
- **Server-Router Distance:** The distance of the connection between the server and router after cabling, which is crucial for fast and stable server operations.
- **Most Popular Printer:** This analysis identifies which printer is the busiest among the options. It assumes that **each student selects the nearest printer to their computer.**

Here are some supplementary instructions:

- Your lab has **a router a server few printers, and several computers.**
- Definition of **Distance**: the cumulative length of the cables that connect them.
- No two devices have the same distance between them.
- The index corresponding to each device is **random**.
- Each student has their own computer. They use the nearest printer, and if two printers are equally close, they choose the one with the **lower device index**.
- If two or more printers have the same amount of usage, return the **smallest index**.



```
import java.util.List;
import java.util.Map;

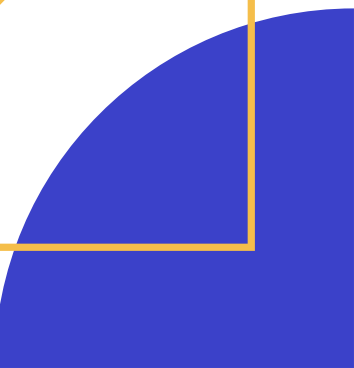
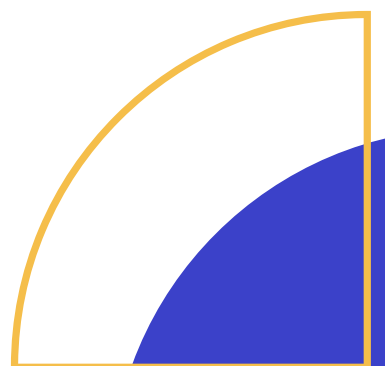
class LabNetworkCabling {

    public LabNetworkCabling(Map<Integer, String> deviceTypes, List<int[]> links){
        // create a Minimum Spanning Tree
    };

    public int cablingCost() {
        int cost = 0;
        // calculate the total cost
        return cost;
    }

    public int serverToRouter(){
        int srDistance = 0;
        // find the path distance between the server and the router
        return srDistance;
    }

    public int mostPopularPrinter(){
        int printerIndex = 0;
        // find the most popular printer and return its index
        return printerIndex;
    }
}
```



```
public static void main(String[] args) {
```

```
// [device index, device type]
```

```
Map<Integer, String> deviceTypes = Map.of(
```

```
    0, "Router",
```

```
    1, "Server",
```

```
    2, "Printer",
```

```
    3, "Printer",
```

```
    4, "Computer",
```

```
    5, "Computer",
```

```
    6, "Computer"
```

```
);
```

```
// [device a, device b, link distance (cable length)]
```

```
List<int[]> links = List.of(
```

```
    new int[]{0, 1, 4},
```

```
    new int[]{1, 2, 2},
```

```
    new int[]{2, 4, 1},
```

```
    new int[]{0, 3, 3},
```

```
    new int[]{1, 3, 8},
```

```
    new int[]{3, 5, 7},
```

```
    new int[]{3, 6, 9},
```

```
    new int[]{0, 6, 5}
```

```
);
```

```
LabNetworkCabling Network = new LabNetworkCabling(deviceTypes, links);
```

```
System.out.println("Total Cabling Cost: " + Network.cablingCost());
```

```
System.out.println("Distance between Server and Router: " + Network.serverToRouter());
```

```
System.out.println("Most Popular Printer: " + Network.mostPopularPrinter());
```

```
}
```

Search ...

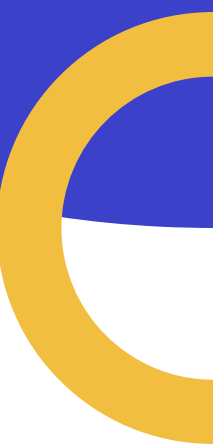


## Expected Output



Total Cabling Cost: 22

Distance between Server and Router: 4

Most Popular Printer: 3



## Hint

- Both the Kruskal and Prim algorithms can be implemented in this assignment. Interested students are encouraged to try both to compare the performance speed of the two Minimum Spanning Trees (MSTs). 
- In the implementation of the MST, you may use the library provided in the course materials. It is advisable to enhance the MST class with some additional functions for better functionality.
- Both **Depth-First Search** and **Breadth-First Search** are valid methods for pathfinding, though other techniques may be suitable as well. If time permits, experimenting with various methods to observe the differences in their performance speeds is recommended. In fact, there is quite a significant difference in speed. 

# TestCase

Time Limit: 720 ms

- $P$  : num of printers
- $C$  : num of computers
- $L$  : num of links
- 20 points:  $P \leq 3, C \leq 3, L \leq 8$
- 20 points:  $P \leq 5, C \leq 5, L \leq 10$  (Special Case)
- 20 points:  $N \leq 20, M \leq 20, L \leq 80$
- 20 points:  $N \leq 400, M \leq 400, L \leq 800$
- 20 points:  $N \leq 2000, M \leq 2000, L \leq 4000$





Search ...



# THANK YOU

**Release date:** 05/20 16:00

**TA hour:** 05/27 13:30

**Due date:** 05/29 21:00

**PDSA HW9**

