GRAPH – LAB 07

EC 4070

DATA STRUCTURES AND ALGORITHMS

NAME : **WIJAYAWARDHANA W.A.H.A.**

REGISTRATION NO. : **2019/E/166**

SEMESTER : **SEMESTER 04**

DATE ASSIGNED : **30 MARCH 2022**

01.

Code:

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

/**
 *
 * @author 2019e166
 */
public class JungleRun_2019_E_166_L7 {
   int mapSize;
   Scanner scanner = new Scanner(System.in);
   int currentPlaceIndex01 = 0;
   int currentPlaceIndex02 = 0;
   ArrayList<String> pathElementList = new ArrayList<>();
   String[][] mapElements = setMapElements(mapSize);

   /**
    * For set mapSize variable and calling setMapElements.
    */
   public void setMapSize()
   {
      System.out.println("Enter the map size : ");
      mapSize = scanner.nextInt();
      mapElements = setMapElements(mapSize);
      findPath();

   }

   /**
    * setMapElements use for set elements into 2D array.
    * @param mapSize
    * @return
    */
   public String[][] setMapElements(int mapSize)
   {
      String[][] mapElementsN = new String[mapSize][mapSize];
      for(int i =0; i < mapSize; i++)
      {
         for(int j =0; j<mapSize; j++)
         {

            mapElementsN[i][j] = scanner.next();
         }
      }
```

```
        return mapElementsN;
    }

    /**
     * findPath method use to find the shorted path of the jungle.
     */
    public void findPath()
    {

        boolean north = false;
        boolean east = false;
        boolean west = false;
        boolean south = false;
        if((currentPlaceIndex01-
1>=0)&&(currentPlaceIndex02<mapSize)&&("E".equals(mapElements[currentPlaceIndex01-
1][currentPlaceIndex02])))
        {
            pathElementList.add("E");
            System.out.println("Path : "+pathElementList);
            System.out.println("Length of path : "+pathElementList.size());
            return;
        }
        else
if((currentPlaceIndex01<mapSize)&&(currentPlaceIndex02+1<mapSize)&&("E".equals(mapEleme
nts[currentPlaceIndex01][currentPlaceIndex02+1])))
        {
            pathElementList.add("E");
            System.out.println("Path : "+pathElementList);
            System.out.println("Length of path : "+pathElementList.size());
            return;
        }
        else if((currentPlaceIndex01<mapSize)&&(currentPlaceIndex02-
1>=0)&&("E".equals(mapElements[currentPlaceIndex01][currentPlaceIndex02-1])))
        {
            pathElementList.add("E");
            System.out.println("Path : "+pathElementList);
            System.out.println("Length of path : "+pathElementList.size());
            return;
        }
        else
if((currentPlaceIndex01+1<mapSize)&&(currentPlaceIndex02<mapSize)&&("E".equals(mapEleme
nts[currentPlaceIndex01+1][currentPlaceIndex02])))
        {
            pathElementList.add("E");
            System.out.println("Path : "+pathElementList);
            System.out.println("Length of path : "+pathElementList.size());
            return;
        }
```

```
        else
if((currentPlaceIndex01+1<mapSize)&&(currentPlaceIndex02<mapSize)&&("P".equals(mapEleme
nts[currentPlaceIndex01+1][currentPlaceIndex02])))
        {
            south = true;
            pathElementList.add("P");
            currentPlaceIndex01++;
            findPath();
        }
        else
if((currentPlaceIndex01<mapSize)&&(currentPlaceIndex02+1<mapSize)&&("P".equals(mapEleme
nts[currentPlaceIndex01][currentPlaceIndex02+1])))
        {
            east = true;
            pathElementList.add("P");
            currentPlaceIndex02++;
            findPath();
        }
        else if((currentPlaceIndex01-
1>=0)&&(currentPlaceIndex02<mapSize)&&("P".equals(mapElements[currentPlaceIndex01-
1][currentPlaceIndex02])))
        {
            north = true;
            pathElementList.add("P");
            currentPlaceIndex01--;
            findPath();
        }
        else if((currentPlaceIndex01<mapSize)&&(currentPlaceIndex02-
1>=0)&&("P".equals(mapElements[currentPlaceIndex01][currentPlaceIndex02-1])))
        {
            west = true;
            pathElementList.add("P");
            currentPlaceIndex02--;
            findPath();
        }
    }


    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        JungleRun_2019_E_166_L7 newObject = new JungleRun_2019_E_166_L7();
        newObject.setMapSize();
    }

}
```

Output:

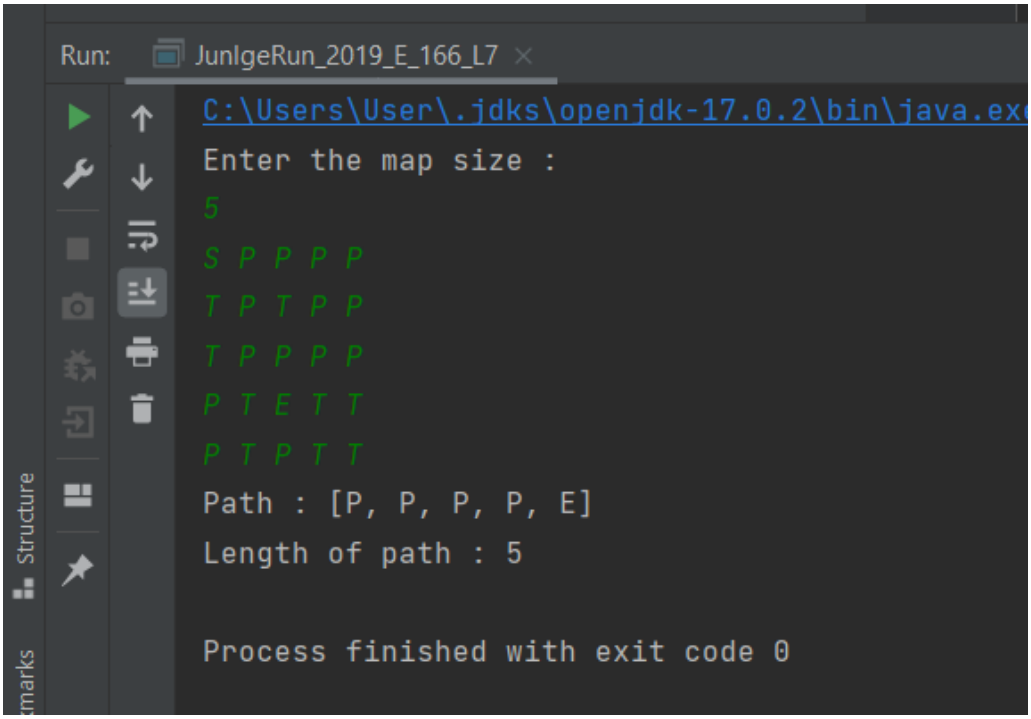

FIGURE 01 - OUTPUT