Name :- Dhanraj Wanjare

Email. – drwanjareafnt@gmail.com

Contact.- 8989413284

Ques 1).

Program of Salesman Problem:-
Using IDE - Eclipse.

```
import java.awt.List;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collection;
import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Scanner;
import java.util.StringTokenizer;

public class SalesManProblem{

    public static void main(String[] args) throws FileNotFoundException {

        Scanner in = new Scanner(new FileInputStream("D:/real.txt"));// to read

                                // data

                                // from

                                // file
        int[][] arr = new int[50][50];// defining 50*50 matrix because 50 cities

        int k = 0;
        while (in.hasNextLine()) {

            String line = in.nextLine();

            String[] str = line.split("\\s");

            int j = 0;

            for (String s : str) {
                int value = Integer.parseInt(s.trim());
                arr[k][j++] = value;// storing data from file to matrix
                arr[j - 1][k] = value;

            }

            k++;

        }


        System.out.println("After storing data from file to Matrix");
        for (int[] a : arr) {
            for (int i : a) {
                System.out.print(i + "\t");// displaying matrix after stroring
                                                            // data from
file to matrix
            }
            System.out.println();
        }

        LinkedList<Integer> ll = new LinkedList<Integer>();// to store root
        boolean list_flag;// if it has true that means need to iterate once
                                    // again otherwise no need to continue the
process

        while (true) {
            list_flag = false;

            ArrayList<Integer> list = new ArrayList<Integer>();// for second

                // density
            ArrayList<Integer> list2 = new ArrayList<Integer>();// for min

                // eleements
            int min1 = 99999;
            int min2 = 99999;
            for (int i = 0; i < arr.length; i++) {
                min1 = 999999;
                min2 = 999999;
                for (int j = 0; j < arr[i].length; j++) {
                    if (arr[i][j] < min1) {
                        min2 = min1;
                        min1 = arr[i][j];
                    } else if ((min1 == arr[i][j]) && (min1 != min2)) {
                        min2 = min1;
                        // System.out.println("min2= " + i);
                    } else if ((arr[i][j] < min2) && (arr[i][j]) != min1) {
                        min2 = arr[i][j];
                    }

                }
```

```
	list.add(min2 - min1);// storing second density values
	list2.add(min1);// storing min values

}

Iterator<Integer> list1_Iterator = list2.iterator();
Iterator<Integer> list_Iterator = list.iterator();

/*
 * to check weather next iteration required or not true means
 * another iteration required
 */
while ((list_Iterator.hasNext()) && (list1_Iterator.hasNext())) {
	int ele1 = list_Iterator.next();
	int ele2 = list1_Iterator.next();

	if ((ele1 <= 999) && (ele2 != 9999)) {
		list_flag = true;
		break;
	}

}

/*
 * next iteration not required but connect everyone to min cell
 * enries when remaining min elements<9999
 */
if (!list_flag) {
	System.out.println("final");
	Iterator<Integer> i = list2.iterator();
	Iterator<Integer> ii = list.iterator();
	while (i.hasNext() && ii.hasNext()) {
		int ee = i.next();

		if (ee != 9999) {

			int c = list2.indexOf(ee);

			int m = arr[c][0];
			int m_index = 0;
			for (int kk = 1; kk < arr[c].length; kk++) {
				if (arr[c][kk] < m) {
					m = arr[c][kk];
					m_index = kk;
				}

			}

			if (ll.contains(c)) {
				if (!ll.contains(m_index))
					ll.add(ll.indexOf(c) + 1, m_index);
```

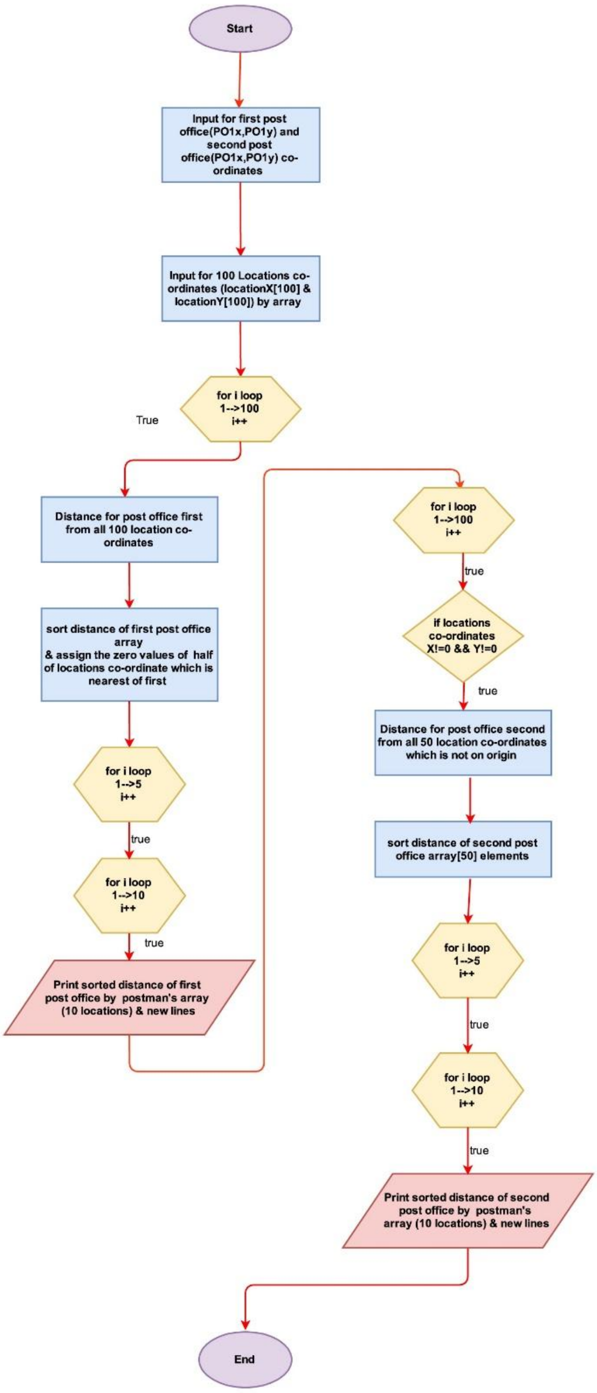
```
                }

            }

        }

        break;
    }
    int row = list.indexOf(Collections.max(list));// finding row that is

    // with Max Density

    // value

    int min = list2.get(row);// min value in that max density row

    // finding the index of min value of density row
    int index = 0;
    for (int i = 0; i < arr[row].length; i++) {
        if (arr[row][i] == min) {
            index = i;
            break;
        }
    }

    // adding the path in linkedlist
    ll.add(row);
    ll.add(index);

    // modifieng the corresponding row values to 9999
    for (int i = 0; i < arr[row].length; i++) {
        arr[row][i] = 9999;
    }

    // modifieng the corresponding column values to 9999
    for (int i = 0; i < arr.length; i++) {
        arr[i][index] = 9999;
    }

    // changinf the a[j][i] to 9999
    arr[index][row] = 9999;

}
// Matrix after final iteration
System.out.println("Final Matrix something like---");
for (int[] a : arr) {
    for (int i : a) {
        System.out.print(i + "\t");
    }
    System.out.println();
}
```

```
        // deleting the locations that are in adjacent in path
        for (int i = 2; i < ll.size() - 1; i++) {
            if (ll.get(i) == ll.get(i + 1))
                ll.remove(i);
        }

        System.out.println("final path=");
        Iterator<Integer> ll_i = ll.iterator();
        while (ll_i.hasNext())
        {
            System.out.print(ll_i.next() + "->");
        }


        in.close();

    }

}
```

Question 2)

Flow Chart Of Post Office Problem

09/12/2016 Preview


Start
Input for first post office(PO1x,PO1y) and second post office(PO1x,PO1y) co-ordinates
Input for 100 Locations co-ordinates (locationX[100] & locationY[100]) by array
for i loop 1-->100 i++
True
Distance for post office first from all 100 location co-ordinates
sort distance of first post office array & assign the zero values of half of locations co-ordinate which is nearest of first
for i loop 1-->5 i++
true
for i loop 1-->10 i++
true
Print sorted distance of first post office by postman's array (10 locations) & new lines
for i loop 1-->100 i++
true
if locations co-ordinates X!=0 && Y!=0
true
Distance for post office second from all 50 location co-ordinates which is not on origin
sort distance of second post office array[50] elements
for i loop 1-->5 i++
true
for i loop 1-->10 i++
true
Print sorted distance of second post office by postman's array (10 locations) & new lines
End

2/3

Program For Post Office Code and Approach by comments line
Written on Notepad and run on console

```
 import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;
import java.util.Arrays;


class PostOffice
{
    public static void main(String[] args)
    {
        //PostOffice po = new PostOffice();
        Scanner scan = new Scanner(System.in);

        System.out.println("Enter the point of first location ");
        int postOffice1_x=scan.nextInt();          //location point 1 of post_office first
        int postOffice1_y=scan.nextInt();          //location point 2 of post_office first

        System.out.println("Enter the point of second location ");
        int postOffice2_x=scan.nextInt();          //location point 1 of post_office second
        int postOffice2_y=scan.nextInt();          //location point 2 of post_office second

        int[] locationX = new int[100];     //location points Matrices of 100 Rows(100 city's places) and 2 coloumn(co-ordinat x axis)
        int[] locationY = new int[100];     //location points Matrices of 100 Rows(100 city's places) and 2 coloumn(co-ordinat y axis)
        for(int i=0;i<100;i++)
        {
            locationX[i]=scan.nextInt();     //100 locations point Scan (X  axis co-ordinate) point from console
            locationY[i]=scan.nextInt();     //100 locations point Scan (Y  axis co-ordinate) point from console
        }

        int[] distanceFromFirstPostOffice1 = new int[100];  //object for First postoffice object array
        int[] sortdistanceFromFirstPostOffice1 = new int[100];  //object for First sort postoffice object array

        for(int i=0;i<100;i++)
        {
```

```
      distanceFromFirstPostOffice2[i]= Math.sqrt((postOffice1_x-
locationX[i])*(postOffice1_x-locationX[i])+(postOffice1_y-locationY[i])*(postOffice1_y-
locationY[i]));
      //Diststance from first post Office
      sortdistanceFromFirstPostOffice1[i]=i; //stroring index for location ponts which
sorted we will assign with zero.

    }
    int temp,k=0;
    //Arrays.sort(distanceFromFirstPostOffice1);
    for(int i=0;i<99;i++)
    {
      min=i;
      for(int j=i+1;j<100;j++)   //sort the distance of first nearest postman places
      {
        if(distanceFromFirstPostOffice1[min]>distanceFromFirstPostOffice1[j])
         {
          min=j;
          sortdistanceFromFirstPostOffice1[k++]=j;
         }
      }
      if(min!=i)
      {
        temp=distanceFromFirstPostOffice1[i];
        distanceFromFirstPostOffice1[i]=distanceFromFirstPostOffice1[min];
        distanceFromFirstPostOffice1[min]=temp;
      }
    }
    for(int i=0;i<k;i++) //indexes assighn by zero hwose used
    {
      locationX[locationX[i]]=0;
      locationY[locationX[i]]=0;
    }
    //int k=0;
    for(int i=0;i<5;i++)
    {
      for(int j=0;j<10;j++)
        System.out.println(distanceFromFirstPostOffice1[j]+" "); //postoffice point whch is
nearest of first postoffice postman's
      printf("\n");
    }
    int[] distanceFromFirstPostOffice2 = new int[50];  //object for second postoffice object
array

    for(int i=0;i<50;i++)
    {
      if(locationX[i]==0 && locationY[i]==0)
      {
        distanceFromFirstPostOffice1[i]= Math.sqrt((postOffice2_x-locationX[i])*(postOffice2_x-locationX[i])+(postOffice2_y-locationY[i])*(postOffice2_y-locationY[i]));
        //Diststance from first post Office
      }

    }

    Arrays.sort(distanceFromFirstPostOffice2);


    //int k=0;
    for(int i=0;i<5;i++)
    {
      for(int j=0;j<10;j++)
        System.out.println(distanceFromFirstPostOffice2[j]+" "); //postoffice point whch is nearest of second postoffice postman's
      printf("\n");
    }

  }
}
```

Thanking You.
Regards
Dhanraj Wanjare