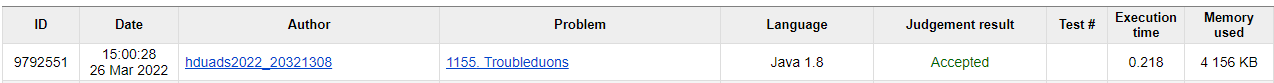
Laboratory work #1

Student: CAO Xinyang  
Student ID: 20321308  
Timus Name: hduads2022\_20321308

Mail: c.x\_yang@foxmail.com

Problem #1155

Screenshot from Timus:



Explanation of algorithm:

We can divide points into two groups, ACFH and BDEG. Points in the same group can be transferred to each other. So we transfer all the points of ACFH to A, all the points of BDEG to B, and then eliminate AB. Before we do that, however, we need to traverse and eliminate each edge first to reduce the number of transitions because the number of operations we can perform is limited (less than 1000).

Computational complexity of algorithm:

The complexity of the code is linearly related to each number N input.

T(N) = O(N)

Source code:

import java.util.Scanner;

import java.util.stream.Stream;

public class App {

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

Scanner scan = new Scanner(System.in);

String input = scan.nextLine();

String[] arr = input.split(" ");

Integer[] troubleduons = Stream.of(arr).mapToInt(Integer::parseInt).boxed().toArray(Integer[]::new);

int sum1 = troubleduons[0] + troubleduons[2] + troubleduons[5] + troubleduons[7];

int sum2 = troubleduons[1] + troubleduons[3] + troubleduons[4] + troubleduons[6];

if(sum1 != sum2){

System.out.println("IMPOSSIBLE");

}else{

while(troubleduons[0] != 0 && troubleduons[1] != 0){

troubleduons[0] --;

troubleduons[1] --;

System.out.println("AB-");

}

while(troubleduons[1] != 0 && troubleduons[2] != 0){

troubleduons[1] --;

troubleduons[2] --;

System.out.println("BC-");

}

while(troubleduons[2] != 0 && troubleduons[3] != 0){

troubleduons[2] --;

troubleduons[3] --;

System.out.println("CD-");

}

while(troubleduons[0] != 0 && troubleduons[3] != 0){

troubleduons[0] --;

troubleduons[3] --;

System.out.println("AD-");

}

while(troubleduons[0] != 0 && troubleduons[4] != 0){

troubleduons[0] --;

troubleduons[4] --;

System.out.println("AE-");

}

while(troubleduons[1] != 0 && troubleduons[5] != 0){

troubleduons[1] --;

troubleduons[5] --;

System.out.println("BF-");

}

while(troubleduons[2] != 0 && troubleduons[6] != 0){

troubleduons[2] --;

troubleduons[6] --;

System.out.println("CG-");

}

while(troubleduons[3] != 0 && troubleduons[7] != 0){

troubleduons[3] --;

troubleduons[7] --;

System.out.println("DH-");

}

while(troubleduons[4] != 0 && troubleduons[5] != 0){

troubleduons[4] --;

troubleduons[5] --;

System.out.println("EF-");

}

while(troubleduons[5] != 0 && troubleduons[6] != 0){

troubleduons[5] --;

troubleduons[6] --;

System.out.println("FG-");

}

while(troubleduons[6] != 0 && troubleduons[7] != 0){

troubleduons[6] --;

troubleduons[7] --;

System.out.println("GH-");

}

while(troubleduons[4] != 0 && troubleduons[7] != 0){

troubleduons[4] --;

troubleduons[7] --;

System.out.println("EH-");

}

while(troubleduons[2] != 0){

troubleduons[2] --;

troubleduons[0] ++;

System.out.println("AD+");

System.out.println("CD-");

}

while(troubleduons[3] != 0){

troubleduons[3] --;

troubleduons[1] ++;

System.out.println("BC+");

System.out.println("CD-");

}

while(troubleduons[4] != 0){

troubleduons[4] --;

troubleduons[1] ++;

System.out.println("BF+");

System.out.println("EF-");

}

while(troubleduons[5] != 0){

troubleduons[5] --;

troubleduons[0] ++;

System.out.println("AE+");

System.out.println("EF-");

}

while(troubleduons[6] != 0){

troubleduons[6] --;

troubleduons[1] ++;

System.out.println("BF+");

System.out.println("FG-");

}

while(troubleduons[7] != 0){

troubleduons[7] --;

troubleduons[0] ++;

System.out.println("AE+");

System.out.println("EH-");

}

while(troubleduons[0] != 0){

troubleduons[0] --;

System.out.println("AB-");

}

}

scan.close();

}

}