Зуйкевич Лидия, 7 группа

Лабораторная работа №3

Вариант 7

**Задача:**

**Задача 7. Олимпиада**

На олимпиаду прибыли N человек. Некоторые из них знакомы между собой. Круг знакомств задан матрицей A размера N ×N. Элемент матрицы равен 1, если человек i знаком 4 с человеком j (если i-й человек знает j-го, то это значит, что j-й человек знает i-го), и элемент матрицы равен 0, если i-й человек не знаком с человеком j. Необходимо определить, можно ли опосредованно (незнакомые люди могут познакомиться только через общего знакомого) перезнакомить всех между собой.

**Формат входных данных:**

Первая строка содержит число N людей (2 ≤ N ≤ 500). Затем идут N строк файла, которые задают матрицу знакомств A (каждой строке матрицы соответствует отдельная строка входного файла).

**Формат выходных данных:**

Если можно опосредованно перезнакомить всех прибывших на олимпиаду между собой, то первая строка файла будет содержать сообщение YES; в противном случае строка файла должна содержать сообщение NO.

**Пример**

|  |  |
| --- | --- |
| **Входной файл** | **Выходной файл** |
| 5  0 1 1 0 0  1 0 1 0 1  1 1 0 1 0  0 0 1 0 0  0 1 0 0 0 | YES |
| 4  0 1 0 0  1 0 0 0  0 0 0 1  0 0 1 0 | NO |
| 2  0 1  1 0 | YES |

**Построение модели:**

По условию задачи имеем граф, заданный матрицей смежности. Необходимо ответить на вопрос, можно ли перезнакомить всех участников олимпиады между собой при условии, что незнакомые люди могут познакомиться только через общего знакомого. Для этого каждая вершина графа должна быть соединена с любой другой вершиной маршрутом (в условиях задачи это цепочка знакомых людей), т.е. граф должен быть связным. Для проверки графа на связность используем поиск в ширину: если все вершины были достигнуты при первом обходе графа из начальной вершины, то граф содержит 1 компоненту связности, т.е. является связным. Также поскольку не может быть ситуации, когда человек i знаком с человеком j, а человек j с человеком i – нет, то граф является неориентированным, поэтому можем рассматривать только правый верхний треугольник матрицы смежности, т.к. она симметричная.

**Листинг программы:**

import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.LinkedList;  
import java.util.Queue;  
import java.util.StringTokenizer;  
  
public class BFS {  
 private int n;  
 private int[][] matr;  
 private int[] is\_visited;  
 private Queue<Integer> queue;  
 private int len;  
  
 BFS(String filename) throws IOException{  
 BufferedReader br = new BufferedReader(new FileReader(filename));  
 StringTokenizer st = new StringTokenizer(br.readLine(), " ");  
  
 this.n = Integer.*parseInt*(st.nextToken());  
 this.matr = new int[n][n];  
 this.is\_visited = new int[n];  
 this.queue = new LinkedList<>();  
  
 for(int i = 0; i < n; i++){  
 st = new StringTokenizer(br.readLine(), " ");  
 for(int j = 0; j < n; j++){  
 matr[i][j] = Integer.*parseInt*(st.nextToken());  
 }  
 }  
 this.len = 1;  
  
  
 }  
  
 public void BreadthFirstSearch(int start){  
 queue.add(start);  
 is\_visited[start] = 1;  
 int p;  
  
 while(!queue.isEmpty()){  
 p = queue.poll();  
  
 for(int j = p; j < n; j++){  
 if(matr[p][j] == 1 && is\_visited[j] == 0){  
 is\_visited[j] = 1;  
 queue.add(j);  
 }  
 }  
 }  
  
  
 }  
 public void solve() throws IOException{  
 FileWriter fw = new FileWriter("output.txt");  
  
 this.BreadthFirstSearch(0);  
 for(int i = 0; i < n; i++){  
 if(is\_visited[i] == 0){  
 fw.write("NO");  
 fw.close();  
 return;  
 }  
 }  
  
 fw.write("YES");  
 fw.close();  
 }  
 public static void main(String[] args) throws IOException{  
 BFS obj = new BFS("input.txt");  
 obj.solve();  
 }  
}

**Вывод программы:**

Проверим работу программы на примерах из условия задачи:



 

 