

Kuis Lab 2 - Struktur Data dan Algoritma
Kamis, 13 Desember 2018
Pukul 18.30 - 20.30 WIB

Peraturan Quiz:

1. Peserta kuliah menyelesaikan quiz sendiri, tanpa bantuan siapapun. Selama pengerjaan quiz, peserta kuliah hanya diperbolehkan mengakses Scele, Aren, dan Java API.
2. Peserta kuliah mengerjakan quiz hanya dengan menggunakan komputer di lab. Tidak boleh menggunakan laptop sendiri.
3. Selama kuis, alat komunikasi dimasukkan ke dalam tas. Tas dikumpulkan di depan sebelum quiz dimulai.
4. Tidak ada perpanjangan waktu pengumpulan.
5. Ingat bahwa nilai dari grader bukan merupakan nilai akhir Anda. Nilai di Aren hanya sebagai alat bantu estimasi nilai mentah Anda.
6. Segala bentuk **PLAGIARISME** dan pelanggaran integritas akan memperoleh sanksi tegas

Soal

Diberikan ***N*** buah ***undirected graph***, buatlah sebuah *method* untuk mendeteksi apakah pada graf tersebut terdapat *cycle* (siklis) atau tidak.

Siklis adalah sirkuit dengan panjang ≥ 3

Format Input

1. Baris pertama adalah *N*, merupakan jumlah graf input
2. Baris kedua terdiri atas 2 buah bilangan bulat *A* dan *B* dipisahkan dengan spasi. *A* merupakan jumlah vertex, *B* merupakan jumlah edge.
3. *B* baris berikutnya merupakan *list of edge*. Pada masing-masing baris terdapat dua bilangan bulat *X* dan *Y*, menandakan vertex *X* terhubung dengan vertex *Y*
4. Poin 2 dan 3 diulang sebanyak *N* buah graf.

Format Output

N buah baris yang merupakan jawaban apakah *N* buah graf terdapat *cycle* atau tidak (lihat contoh input-output)

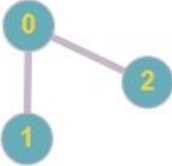
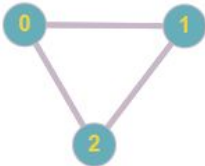
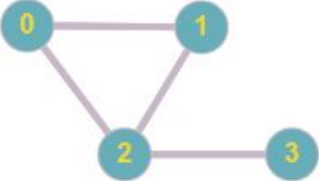
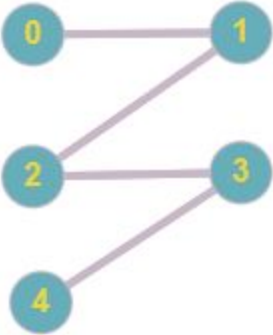
Batasan Input

Graf yang diberikan sebagai input soal adalah sebuah **graf tidak kosong** (jumlah vertexes sekurang-kurangnya 1)

$$1 \leq \text{jumlah vertex} \leq 20.000$$

$$0 \leq \text{jumlah edge} \leq 20.000$$

Contoh

Input	Representasi Graf	Output
1 3 2 0 1 0 2		false
1 3 3 0 1 0 2 1 2		true
2 4 4 0 1 0 2 1 2 2 3 5 4 0 1 1 2 2 3 3 4	<p>Graf 1</p>  <p>Graf 2</p> 	true false

Template Code

Representasi graf yang disediakan pada template berikut adalah adjacency list. Disediakan juga method untuk membuat representasi adjacency matrix. Silakan pilih salah satu.

Anda juga tidak harus menggunakan template code ini. Anda boleh menggunakan representasi graf Anda sendiri.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.util.Iterator; import java.util.ArrayList;

/* Class Graph memiliki properti int JUMLAH_VERTEX, merupakan jumlah vertex pada graf
 *
 * Vertex pada graf dilabeli menggunakan bilangan bulat 0 sampai dengan JUMLAH_VERTEX-1.
 *
 * Adjacency List direpresentasikan menggunakan array of ArrayList. Index pada array
 * merepresentasikan vertex i. ArrayList berisi bilangan-bilangan bulat, merepresentasikan
 * vertex yang terhubung dengan vertex i.
 *
 *
 * Contoh: Berikut ini merupakan representasi graf dengan 4 buah vertex
 * (3) (0) | array[0] => 1 2
 * \ / \ | array[1] => 0 2
 * \ / \ | array[2] => 0 1 3
 * (2)----- (1) | array[3] => 2
 */

class Graph{
    private static int JUMLAH_VERTEX;
    private static ArrayList<Integer> adjList[];
    private static boolean visited[];
    private boolean[][] adjacencyMatrix;

    Graph(int v){
        JUMLAH_VERTEX = v;
        adjList = new ArrayList[v];
        visited = new boolean[v];
        for (int i=0; i<JUMLAH_VERTEX; i++){
            adjList[i] = new ArrayList();
            visited[i] = false;
        }
    }

    void adjMatrixOf(){
        boolean[][] adjacencyMatrix = new boolean[JUMLAH_VERTEX][JUMLAH_VERTEX];
        for(int i=0; i<JUMLAH_VERTEX; i++){
            for(int j=0; j<JUMLAH_VERTEX; j++){
                adjacencyMatrix[i][j]=true;
                adjacencyMatrix[j][i]=true;
            }
        }
    }

    int getJmlVertex(){
        return JUMLAH_VERTEX;
    }
}
```

```

    void addEdge(int v, int w){
        adjList[v].add(w);
        adjList[w].add(v);
    }

    boolean detectCycle(){
        //implementasi program Anda
    }

    /* Method untuk mencetak representasi adjacency list dari sebuah graf (untuk gambaran
saja) */
    void printGraph(){
        for(int i=0; i<JUMLAH_VERTEX; i++){
            Iterator<Integer> itr = adjList[i].listIterator();
            System.out.print(i+" => ");
            while(itr.hasNext()){
                System.out.print(itr.next()+" ");
            }
            System.out.println("");
        }
    }

class GraphDemo{
    public static void main(String args[]) throws IOException{
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

        int jmlGraf = Integer.parseInt(reader.readLine());

        String[] in;
        int jmlVertex;
        int jmlEdge;

        Graph graf;

        boolean[] out = new boolean[jmlGraf];

        for(int x=0; x<jmlGraf; x++){
            //Silakan modifikasi jika diperlukan
            in = reader.readLine().split(" ");
            jmlVertex = Integer.parseInt(in[0]);
            jmlEdge = Integer.parseInt(in[1]);

            graf = new Graph(jmlVertex);

            for(int j=0; j<jmlEdge; j++){
                in = reader.readLine().split(" ");
                int a = Integer.parseInt(in[0]);
                int b = Integer.parseInt(in[1]);
                graf.addEdge(a, b);
            }

            out[x] = graf.detectCycle();
        }
        for(boolean i : out){
            System.out.println(i);
        }
    }
}

```