

Academiejaar: 2016–2017
Opleiding: Bachelor in de Toegepaste informatica
Fase: 1
Examinator: G. Jongen, M. Lens, J. Van Hee
OPO: MBI72 – Bomen & Grafen
OLA: MBI72a – Bomen & Grafen
Hulpmiddelen: Eigen laptop, wifi netwerk xtoledo(5G)
Datum: 13 juni 2017
Beginuur: 13
Tijdsduur: 3 uur



Student: Nr: Reeks:

Deze opgaven vormen het tweede deel van het examen. Je krijgt het pas na het inleveren van het schriftelijk deel. Beide opgaves programmeer je op je eigen laptop. Je mag hierbij alles gebruiken wat op je laptop staat.

Je mag enkel verbinding maken met het Wifi-netwerk “xtoledo” of “xtoledo5G”. Als je dat laatste ziet in de lijst met netwerken, kan je best hiermee verbinding maken. Het wachtwoord is “bomenengrafen”.

Dit is het enige toegelaten netwerk. Er is enkel toegang tot x.toledo.ucll.be. We gebruiken de xtoledo omgeving enkel om de code van het examen af te geven.

Doorloop nu volgende stappen:

1. Voor dit examen maak je een nieuw java-project aan in Eclipse. De naam van je project is “1TXreeksnummer_FamilienaamVoornaam”, waarbij je vanzelfsprekend je reeksnummer (een getal van 1 tot 12) invult en je eigen naam (graag eerst de familienaam).
2. Je maakt in je src map een package domain en een package ui aan. Elke examenvraag (van de twee) geeft een nieuwe klasse in domain en een bijhorende driverklasse in ui. Je project zal dus bestaan uit twee javabestanden in domain en twee javabestanden in ui.
3. Als je klaar bent, exporteer je je project in Eclipse als een zip bestand (klik in de Package Explorer met de rechtermuisknop op je project en kies export enz). Geef de zip dezelfde naam als je project (dus “1TXreeksnummer_FamilienaamVoornaam”). Als je een lege zip afgeeft, kan je ook geen punten krijgen. *Controleer m.a.w. of je zip de juiste bestanden bevat (door het bestand eens te onzippen en te kijken naar de inhoud).*
4. Log in op x.toledo.ucll.be. Je vindt er het examenvak “Bomen en Grafen - examen 13 juni 2017”.
5. Upload je oplossing (geëxporteerde zip met de juiste naam) naar xToledo. Sluit je computer nog niet af, maar vraag aan een toezichter om na te kijken of er onder jouw naam een bestand is geüploadet.
6. Na de controle en bevestiging van de opzichter dien je je examen officieel in via de knop “verlaat examen”, links onder het cursusmenu.

Veel succes!

Vraag 1. Binaire Bomen (5 punten)

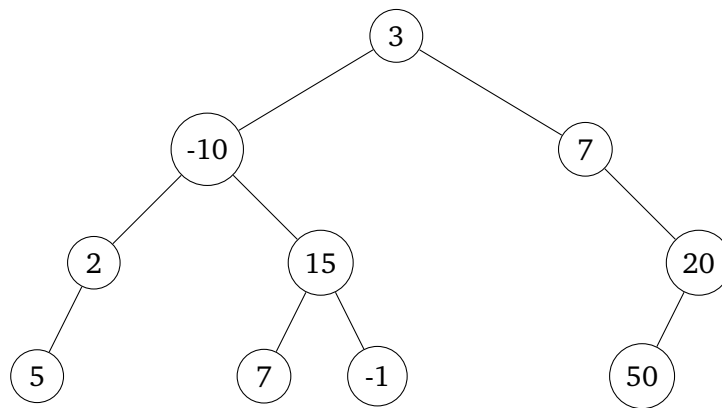
Gegeven een binaire boom met gehele getallen als waarde voor de knooppunten. Definieer hiervoor – naar analogie met de `BinaryTree<E>` klasse die we gebruikten in de oefenzitting – een nieuwe klasse `BinaryTreeInt` die bestaat uit een data veld van het type `Integer` en een `leftTree` en `rightTree` van het type `BinaryTreeInt`. Dat ziet er bvb. zo uit:

Listing 1 Klasse `BinaryTreeInt`

```
public class BinaryTreeInt {  
    private Integer data;  
    private BinaryTreeInt leftTree, rightTree;  
  
    //vul verder zelf aan met constructoren en methode kinderSom()  
}
```

Schrijf een methode `kinderSom()` die de boom doorloopt en de som berekent van alle knopen waarvan de ouderknoop een veelvoud van 5 is. Als je bij wijze van voorbeeld deze methode toepast op de binaire boom uit figuur 1 bekom je als resultaat 73.

Maak een `BinaryTreeIntDriver` klasse waar je in de main methode de boom uit figuur 1 definieert en het resultaat van de methode `kinderSom()` op deze binaire boom in de console weergeeft.



Figuur 1 Deze binaire boom heeft `kinderSom` gelijk aan 73

Vraag 2. Grafen (5 punten)

Voor deze vraag maak je in je examenproject in Eclipse een nieuwe klasse Graph aan in het package domain. Je kopieert in het bestand Graph.java wat je nodig hebt uit de gelijknamige klasse die we in les 8 gebruikten. Die code begint als volgt:

Listing 2 Klasse Graph

```
public class Graph {
    private final boolean[][] verbindingsMatrix;
    public static final int infty = Integer.MAX_VALUE;

    public Graph(int[][] matrix) {
        if (!isGeldigeVerbindingsMatrix(matrix))
            throw new IllegalArgumentException("No valid nabijheidsmatrix");

        this.verbindingsMatrix = new boolean[matrix.length][matrix.length];
        for (int i = 0; i < matrix.length; i++)
            for (int j = 0; j < matrix.length; j++)
                this.verbindingsMatrix[i][j] = matrix[i][j] == 1;
    }

    private boolean isGeldigeVerbindingsMatrix(int[][] matrix) {
        if (matrix == null || matrix.length != matrix[0].length)
            return false;

        for (int i = 0; i < matrix.length; i++)
            if (matrix[i][i] != 0)
                return false;

        for (int i = 0; i < matrix.length; i++)
            for (int j = 0; j < matrix.length; j++)
                if (matrix[i][j] != 0 && matrix[i][j] != 1)
                    return false;
        return true;
    }

    private int getAantalKnopen() {
        return this.verbindingsMatrix.length;
    }
}
```

De **opgave** is: Schrijf in deze klasse een methode isPad(List<Integer> pad) die teruggeeft of de lijst van integers overeenkomt met een pad in de graaf.

Om je methode te testen doe je het volgende. Maak in het package ui een klasse GraphDriver.

Listing 3 Klasse GraphDriver

```
public class GraphDriver {

    public static void main(String[] args) {

        int data[][] = { { 0, 1, 0, 1, 0, 0, 0 },
                        { 0, 0, 1, 0, 0, 0, 0 },
                        { 1, 0, 0, 1, 1, 0, 0 },
                        { 0, 0, 0, 0, 1, 1, 0 },
                        { 0, 1, 0, 0, 0, 0, 1 },
                        { 0, 0, 0, 0, 1, 0, 0 },
                        { 0, 0, 0, 0, 0, 1, 0 } };

        Graph g = new Graph(data);
    }
}
```

```
int[] pad1 = {1,2,3,5,2};
List<Integer> lpad1 = new ArrayList<>();
for (int i : pad1) lpad1.add(i);
System.out.println(lpad1 + " : " + g.isPad(lpad1));

int[] pad2 = {1,2,3,5,3};
List<Integer> lpad2 = new ArrayList<>();
for (int i : pad2) lpad2.add(i);
System.out.println(lpad2 + " : " + g.isPad(lpad2));

List<Integer> lpad3 = new ArrayList<>();
lpad3.add(1);
System.out.println(lpad3 + " : " + g.isPad(lpad3));

List<Integer> lpad4 = new ArrayList<>();
lpad4.add(1);
lpad4.add(8);
lpad4.add(2);
System.out.println(lpad4 + " : " + g.isPad(lpad4));
    }
}
```

De uitvoer in de console zou er als volgt moeten uitzien:

```
[1, 2, 3, 5, 2] :true
[1, 2, 3, 5, 3] : false
[1] : false
[1, 8, 2] : false
```