

Academiejaar: 2019–2020  
Opleiding: Bachelor in de Toegepaste informatica  
Fase: 2  
Examinator: G. Jongen, T. Eversdijk, J. Van Hee  
OPO: MBI72– Bomen & Grafen  
OLA: MBI72a– Bomen & Grafen  
Hulpmiddelen: Eigen laptop, beperkt wifi netwerk xtoledo  
Datum: 23 juni 2020  
Beginuur: 9  
Tijdsduur: 2 uur + evt. 30 % extra tijd



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”. Dit is het enige toegelaten netwerk. Er is enkel toegang tot x.toledo.ucll.be. We gebruiken de xtoledo omgeving enkel om je wat code te bezorgen en de code van je oplossing af te geven.

Doorloop nu volgende stappen:

1. Voor dit examen maak je een nieuw java-project aan in IntelliJ. De naam van je project is “BG\_FamilienaamVoornaam”.
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 IntelliJ als een zip bestand. Geef de zip dezelfde naam als je project (dus “BG\_FamilienaamVoornaam.zip”). 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 downloaden, te onzippen en de inhoud te bekijken).*
4. Log in op x.toledo.ucll.be.  
Je vindt er het examenvak “2020\_2\_11/06/2020\_nm\_MBI72A Bomen en grafen”.
5. Je vindt op xToledo het bestand templates.zip. Deze zip bevat broncode die je van ons krijgt.
6. Dien je examen officieel in via de knop “verlaat examen”, links onder het cursusmenu.
7. *Laat het zip-bestand onaangeroerd op je laptop staan tot na de examens zodat je een bewijs hebt als er iets technisch fout zou lopen.*

Veel succes!

**Vraag 1. Bomen (6 punten)**

Gegeven een binaire boom. Kopieer de domainklasse van les 3 (binaire bomen) naar je examenproject.

- (a) Schrijf de methode `allNodesHaveTwoChildren()`: `boolean`. Deze methode geeft `true` terug als elke interne knoop van de boom juist twee kinderen heeft. In alle andere gevallen geeft de methode `false` terug.
- (b) Schrijf de methode `onlyLeavesOnDepth(int k)`: `boolean`. Deze methode geeft `true` terug als alle knopen van de boom op de gegeven diepte `k` een blaadje zijn.
- (c) Schrijf de methode `isPerfect()`: `boolean`. Deze methode geeft `true` terug als de gegeven boom perfect is. Een boom is perfect als elke interne knoop juist 2 kinderen heeft én alle blaadjes zich op dezelfde diepte bevinden. Maak gebruik van de methodes die je net programmeerde.

**Vraag 2. Grafen (6 punten)**

Een proefopstelling bestaat uit buizen die met mekaar verbonden zijn via emmers. Water kan via een buis langs boven in een emmer lopen. Onderaan de emmer vertrekken opnieuw telkens een of meerdere buizen. Water dat in een emmer loopt, wordt altijd netjes verdeeld over alle vertrekkende buizen. De constructie lekt. Het “lekpercentage” van een buis geeft aan hoeveel procent van het binnenkomende water niet uitmondt in bijhorende emmer.

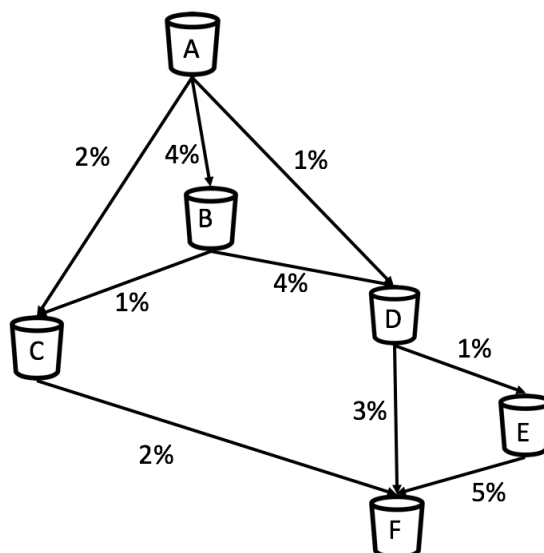


Figure 1: Voorbeeld bij vraag 2

We hebben een programma nodig dat het volgende berekent: gegeven de lekpercentages en de hoeveelheid water die men in de eerste emmer giet, hoeveel water zal uiteindelijk in de laatste emmer terechtkomen?

**Modellering**

We modelleren het probleem als een gerichte graaf. De emmers zijn de knooppunten. De buizen de verbindingen. De lekpercentages zijn de gewichten van de verbindingen. De verbindingsmatrix van voorbeeld 1 is gelijk aan:

```
double data[][]= {
    // A    B    C    D    E    F
    {0.00, 0.96, 0.98, 0.99, 0.00, 0.00 }, //A
    {0.00, 0.00, 0.99, 0.96, 0.00, 0.00 }, //B
    {0.00, 0.00, 0.00, 0.00, 0.00, 0.98 }, //C
    {0.00, 0.00, 0.00, 0.00, 0.99, 0.97 }, //D
    {0.00, 0.00, 0.00, 0.00, 0.00, 0.97 }, //E
    {0.00, 0.00, 0.00, 0.00, 0.00, 0.00}}; //F
```

Je krijgt van ons domain- en ui klasse, zie folder buckets\_gegeven.

Laat je inspireren door het BFS algoritme om dit probleem op te lossen.

Gegeven een binaire boom. Kopieer de domainklasse van les 3 (binaire bomen) naar je examenproject.

**Opdracht**

Schrijf in de domainklasse de methode `buizensysteem(double input, int bron, int put):double`. De veranderlijke `input` is de hoeveelheid water die in de emmer met index `bron` gegoten wordt. De methode berekent de hoeveelheid water die in de emmer met index `put` zal terechtkomen.