

# Gegevensstructuren en Algoritmen: Practicum 1

Academiejaar 2016-2017

## Inhoudsopgave

<b>1 Gedragscode</b>	<b>1</b>
<b>2 Wat is de bedoeling van dit practicum?</b>	<b>2</b>
<b>3 Technische uitwerking</b>	<b>2</b>
<b>4 Criteria</b>	<b>3</b>
<b>5 Deadline</b>	<b>3</b>
<b>6 Communicatie</b>	<b>3</b>
<b>A Hoe Ant Installeren</b>	<b>3</b>
<b>B Hoe Ant Gebruiken</b>	<b>4</b>

## 1 Gedragscode

(Laatste update gedragscode: 5 maart 2015)

De practica worden gequoteerd, en het examenreglement is dan ook van toepassing. Soms is er echter wat onduidelijkheid over wat toegestaan is en niet inzake samenwerking bij opdrachten zoals deze.

De oplossing en/of verslag en/of programmacode die ingediend wordt moet volledig het resultaat zijn van werk dat je zelf gepresteerd hebt. Je mag je werk uiteraard bespreken met andere studenten, in de zin dat je praat over algemene oplossingsmethoden of algoritmen, maar de bespreking mag niet gaan over specifieke code of verslagtekst die je aan het schrijven bent, noch over specifieke resultaten die je wenst in te dienen. Als je het met anderen over je practicum hebt, mag dit er dus NOOIT toe leiden, dat je op om het even welk moment in het bezit bent van een geheel of gedeeltelijke kopie van het opgeloste practicum of verslag van anderen, onafhankelijk van of die code of verslag nu op papier staat of in elektronische vorm beschikbaar is, en onafhankelijk van wie de code of het verslag geschreven heeft (mede-studenten, eventueel uit andere studiejaar, volledige buitenstaanders, internet-bronnen, e.d.). Dit houdt tevens ook in dat er geen enkele geldige reden is om je code of verslag door te geven aan mede-studenten, noch om dit beschikbaar te stellen via publiek bereikbare directories of websites.

Elke student is verantwoordelijk voor de code en het werk dat hij of zij indient. Als tijdens de beoordeling van het practicum er twijfels zijn over het feit of het practicum zelf gemaakt is (bvb. gelijkaardige code, grafieken, of oplossingen met andere practica), zal de student gevraagd worden hiervoor een verklaring te geven. Indien dit de twijfels niet wegwerkt, zal er worden overgegaan tot het melden van een onregelmatigheid, zoals voorzien in het onderwijs- en examenreglement (zie <http://www.kuleuven.be/onderwijs/oer/>).

## 2 Wat is de bedoeling van dit practicum?

In dit practicum doe je hands-on ervaring op over het uitvoeren van een analyse van een algoritme. Dit doe je door een verslag te schrijven dat de onderstaande vraag beantwoordt:

Hoe efficiënt zijn selection sort, insertion sort, en quicksort<sup>1</sup> op random data?

Om deze vraag te beantwoorden, is het noodzakelijk experimenten uit te voeren. Om de scope van dit practicum niet te groot te maken, is het voldoende enkel de volgende experimenten uit te voeren:

- Voer de drie sorteeralgoritmen (selection sort, insertion sort en quicksort) uit op verschillende invoergroottes (vb  $N=1,2,3,\dots,100$ ) en meet het aantal onderlinge vergelijkingen. Plot het resultaat.
- Voer een doubling ratio experiment uit (zoals beschreven in het handboek blz 192 e.v.) voor quicksort en insertion sort. Welk resultaat verwacht je theoretisch? Komt dit overeen met je resultaat? Om te tonen dat je dit goed begrijpt, voorspel je wat de uitvoeringstijd zal zijn van een erg grote invoer (bv. 8 maal groter dan je grootste meting) voor zowel quicksort als insertion sort. Merk op dat voorspellen iets anders is dan meten.
- Welke doubling ratio zou je verwachten bij een  $\sim n^5$  algoritme?



**De nadruk van dit practicum ligt op het verslag.**

Enkele tips: Toon inzicht! Beschrijf niet gewoon wat je ziet, maar verklaar ook waarom! Bestudeer verbanden (eenvoudig voorbeeld: komen je resultaten overeen met de theorie?). Enkel tilde-notatie wordt toegestaan (dus geen grote-O notatie!).

## 3 Technische uitwerking

### 1. Voorbereiding:

- (a) Installeer het programma Ant én voer het eens uit. Doe dit in het begin, zo kom je vlak voor de deadline niet voor verrassingen te staan. Dit document bevat een sectie Hoe Ant Installeren.
- (b) Schrijf JUnit tests voor `QuickSort.java`, `InsertionSort.java`, en `SelectionSort.java`; initieel zullen al je tests dus falen. De bedoeling is dat je JUnit tests testen of je algoritmen wel correct sorteren.

Het commando `ant test` voert jouw eigen tests uit (alle tests met het woord “Test” in de klassenaam) plus een aantal tests die we je gratis geven. De tests die we je geven, testen enkel een paar technische foutjes maar testen niet of je algoritme wel sorteert! Daarvoor met je dus zelf tests schrijven.

Voer regelmatig het commando `ant test` uit zodat je op tijd merkt als je iets fout doet.

- (c) Implementeer `QuickSort.java`, `InsertionSort.java`, en `SelectionSort.java`.
- (d) Debug totdat de tests slagen.
- (e) Als je een main methode wilt schrijven, dan kan dat in `Main.java`. Je kan die main methode uitvoeren met het commando `ant run`.

Als je je main methode via Eclipse wilt uitvoeren, zal je in Eclipse `lib/libpract.jar` moeten toevoegen aan je build path. Je kan zo ook JUnit tests via Eclipse uitvoeren, maar dit voert (typisch) niet alle tests uit. Als je tests uitvoert via Eclipse, voer dan ook nog regelmatig het commando `ant test` uit.

---

<sup>1</sup> We bedoelen de “standaard” versie van QuickSort zoals in het handboek blz 288-294. Gebruik geen initial shuffle.

## 2. Het echte werk

- (a) Voer experimenten uit en schrijf het verslag. Zie sectie Wat is de bedoeling van dit practicum?.

## 3. Releases

- (a) Voer **ant test** uit om je eigen tests uit te voeren en bovendien te controleren op veelvoorkomende fouten.
- (b) Controleer of op je verslag je naam en studentnummer staan zodat we bij het printen weten welk verslag van wie is.
- (c) Maak een zip file met Ant. Deze opgave bevat een sectie Hoe Ant Gebruiken. Ant is verplicht. Zo heeft iedere zip-file dezelfde directory structuur; ervaring leert dat dit niet lukt als studenten de ZIP file met de hand maken. Hierdoor kunnen we sneller verbeteren en dus sneller feedback geven. Als je .zip file duidelijk niet met Ant is gemaakt, zullen we hem niet verbeteren.
- (d) Ant maakt een zip file **build/firstname\_lastname\_studentnumber.zip**. Vul hier je voornaam, achternaam en studentnummer (inclusief letters 'r', 's' of 'm') in in de bestandsnaam. Verwissel niet je voornaam en je achternaam.
  - i. Fout: **Janssens-Jan-r0123456.zip**
  - ii. Fout: **Jan-Janssens-0123456.zip**
- (e) Open je ZIP file en controleer of alles er in zit.
- (f) Upload je ZIP file op Toledo. Je hoeft geen papieren verslag in te dienen.

Indien Toledo down is, mail dan een screenshot hiervan en je zip file naar de verantwoordelijke van dit practicum (zie sectie Communicatie).

## 4 Criteria

Je wordt in de eerste plaats beoordeeld op je verslag. Hiernaast moet je sorteeralgoritmen correct sorteren (dat is verrassend vaak niet het geval) en juist het aantal compares tellen.

## 5 Deadline

De deadline is maandag 27 maart 2017 14:00. Laattijdige inzendingen worden niet aanvaard.

## 6 Communicatie

Stel je vragen via het Toledo forum.

De verantwoordelijke van dit practicum is **jo** punt **vanbulck** apenstaartje **cs** punt **kuleuven** punt **be**. Hier kan je ook naar mailen voor zaken die je niet publiekelijk kan communiceren. Andere practica kunnen andere verantwoordelijken hebben.

## A Hoe Ant Installeren

Ant is niet standaard bijgeleverd bij Java en ook niet bij Windows. Je moet Ant dus eerst installeren.

**Windows thuis:** Het tweede google-resultaat over hoe je Ant installeert onder Windows levert <http://code.google.com/p/winant/> op. Dit is een heel eenvoudige installer. Deze installer vraagt wat de directory is waar JDK is geïnstalleerd; dit is typisch zoiets als `C:\Program Files\Java\jdk1.7.0_17\bin` (afhankelijk van welke JDK je precies hebt).

**Linux thuis:** Voor Ubuntu en Debian: de installatie is eenvoudigweg “`sudo apt-get install ant`” intypen in een terminalvenster. Voor andere distributies: gebruik je package manager.

**PC-labo computerwetenschappen (gebouw 200A):** Ant is reeds geïnstalleerd.

**LUDIT pc-labo:** Ongekend; het is waarschijnlijk veel makkelijker Ant op een eigen machine te installeren.

**Mac OS X:** <sup>2</sup>

1. Open een terminal
2. Type volgende commando's:

```
curl -O http://apache.belnet.be/ant/binaries/apache-ant-1.9.6-bin.zip
unzip apache-ant-1.9.6-bin.zip
sudo mkdir -p /usr/local/
sudo cp -rf apache-ant-1.9.6 /usr/local/apache-ant
export PATH=/usr/local/apache-ant/bin:$PATH
echo 'export PATH=/usr/local/apache-ant/bin:$PATH' >> ~/.profile
```

Ant is nu geïnstalleerd. Als je bij het uitvoeren van “ant” de error krijgt dat je JDK moet installeren, dan moet je dat doen. Je hebt JDK (Java Development Kit) nodig om Java programma's te compileren in het algemeen, dus ook als je Java programma's compileert via Ant.

## B Hoe Ant Gebruiken

1. Start een terminalvenster (dit werkt ook onder Windows: menu start, dan execute, dan “cmd” intypen; zie anders <http://www.google.com/search?q=how+to+open+windows+command>)
2. Navigeer naar de directory waar je bestanden voor dit practicum staan, meer bepaald de directory waar zich `build.xml` in bevindt. Met “cd” verander je van directory en met “ls” (Unix) of “dir” (Windows) toon je de bestanden en directories in de huidige directory.
3. Type “ant release”. Ant doet een aantal checks om je tegen een aantal fouten te beschermen. Check dus of Ant geen error gaf.

---

<sup>2</sup>Bron: <http://gauravstomar.blogspot.be/2011/09/installing-or-upgrading-ant-in-mac-osx.html>