

## **Practicum 2a**

### *Resultaten van studenten sorteren*

Je dient een programma te schrijven waarmee je efficiënt een lijst met studenten kunt laten zien, gesorteerd op bepaalde criteria.

We gaan uit van een lijst met resultaten van een tentamen van een groep studenten, die in verschillende klassen zitten. Om het programma te testen moet je zelf zo'n lijst genereren.

Van een student is het studentnummer, de klas en het cijfer gegeven. De namen van de klassen zijn altijd van de vorm I?2xx, waarbij ? vervangen kan worden door G, N, S of T (de studierichting) en xx vervangen kan worden door 01, 02, .... al naar gelang het aantal klassen dat er is van een bepaalde richting. Het aantal studenten in een klas is altijd maximaal 32.

Het programma moet een aantal studenten aan kunnen maken, standaard is 200, maar je moet dit gemakkelijk kunnen wijzigen in 400 of 800 of meer. Het aantal klassen wordt bepaald door het aantal studenten en de studenten worden evenredig verdeeld over de richtingen en daarom hebben alle richtingen een gelijk aantal klassen. Bij 200 studenten staan er dus 2 klassen per richting klaar, bij 400 studenten 4 klassen, maar bij 800 studenten zouden 7 klassen per richting volstaan - want dan zijn er  $7 \cdot 4 = 28$  klassen met  $28 \cdot 32 = 896$  plaatsen. Het programma moet dit zelf regelen.

Bij het aanmaken krijgt elke student een studentnummer opeenvolgend te beginnen bij 50060001. Het programma wijst een klas toe door steeds één voor één de klassen af te lopen - de eerst vijf studenten zitten dus in klas IG201, IN201, IS201, IT201, IG202, maar een andere volgorde zou ook kunnen en hangt ook van het aantal klassen af. En het programma geeft elke student random een cijfer met 1 decimaal tussen 1,0 en 10. Na het aanmaken moet je de lijst 'schudden', zodat de studentnummers niet meer op volgorde staan. Je kunt daarvoor de methode `shuffle()` gebruiken uit de klasse `StdRandom` die in de library staat.

De aangemaakte lijst van studenten moet gesorteerd kunnen worden en het programma moet de (gesorteerde) lijst kunnen printen. Je moet twee verschillende sorteer functionaliteiten maken en je moet kunnen kiezen hoe je de resultaten wilt sorteren:

1. Je moet de gehele lijst met alle studenten kunnen sorteren op cijfer van hoog naar laag.
2. Je moet de lijst per klas kunnen tonen gesorteerd op studentnummer, en de klassen gesorteerd op alfabet (IG201 bovenaan). Kijk onderaan dit document voor een voorbeeld output van sorteer optie 2.

### **Implementatie**

Denk goed na over de benodigde classes (`Student?`, `ResultList?`, ...) en de datastructuren die je gebruikt. Definieer indien nodig constanten.

Gebruik voor het sorteren op cijfer naar keuze Bubble Sort, Insertion Sort of Selection Sort.

Gebruik voor het groeperen per klas en sorteren op studentnummer een vorm van Bucket Sort. Daarbij zie je de klas dan als Bucket.

## Practicum 2b

Onderzoek de efficiëntie van je programma door verschillende tests uit te voeren. Om de efficiëntie van de beide sorteropties te testen moet je (in een loop) verschillende lijsten van 200, 400, 800, 1600 en 3200 studenten met hun resultaten genereren en steeds de twee bijbehorende gesorteerde lijsten studenten presenteren. Je moet voor beide sorteropties de tests uitvoeren, dus voor het sorteren op cijfer en voor het groeperen per klas met sortering op studentnummer. Je zult dan 10 tests moeten uitvoeren.

Trek conclusies op basis van je tests met betrekking tot de tijd en de ruimte efficiëntie van je implementaties en maak daarbij gebruik van de big O.

In je verslag leg je het programma uit aan de hand van een klassediagram en code snippets. Je laat voorbeeld output zien.

Je laat tabellen en eventueel grafieken zien om de resultaten van de tests te tonen met daarbij een berekening of bepaling van de big O. In de presentatie van les 4 zie je vergelijkbare tabellen en grafieken.

### Voorbeeld output sorteer optie 2

500537860	3,5	IG201
500602044	4,4	IG201
500626056	5,2	IG201
500635805	5,7	IG201
500639561	3,8	IG201
500658043	1,3	IG201
500659869	4,8	IG201
500659986	2,0	IG201
500660108	8,9	IG201
500663514	9,1	IG201
500667087	10,0	IG201
500672461	9,4	IG201
500673014	8,3	IG201
500546088	9,2	IG202
500623980	8,1	IG202
500635137	8,1	IG202
500637844	9,2	IG202
500640510	9,2	IG202
500641310	1,7	IG202
500644291	5,3	IG202
500654777	8,2	IG202
500657412	9,9	IG202
500660614	5,2	IG202