

# Examen Datastructuren en Algoritmen II

*Gelieve elke vraag op een apart dubbel blad te beantwoorden.  
Schrijf op elk blad je naam.*

---

## 1. Theorie : Gedistribueerde algoritmen (4 pten.)

(a) Leg uit waarom het Lehmann–Rabin–algoritme een goed algoritme is.

```
algoritme Lehman-Rabin /* voor het dinerende-filosofen probleem */
    herhaal
n1:   doe private denkactiviteiten tot ge honger krijgt;
t1:   herhaal
t2:   kies at random voor een linkse of rechtse vork;
t3:   wacht tot de gekozen vork vrij is en neem die;
t4:   als de andere vork vrij is neem die en verlaat de p-lus;
t5:   anders leg uw vork neer;
k1:   doe kritische eetactiviteiten;
e1:   leg beide vorken neer;
```

(b) Bij het Peterson–algoritme is wederzijdse uitsluiting verzekerd. Hoe kan dat bewezen worden? (het volstaat de grote lijnen van het bewijs te schetsen)

```
algoritme peterson2P /* voor 2 concurrente processen
                    proces[i] met i = 1..2 */
/* gemeenschappelijke veranderlijken */
    integer beurt = 1; integer vlag[i] = 0;
/* protocol voor proces[i] */
n1:   niet_kritisch_werk();
t1:   vlag[i] = 1;
t2:   beurt = i;
t3:   for(;;) {
t4:   if (vlag[niet_i] < 1 ) break;
t5:   if (beurt != i ) break;}
k1:   kritisch_werk();
e1:   vlag[i] = 0;
```

## 2. Theorie : Splay-bomen (4 pten.)

- (a) Bespreek de *splay*-bewerking in de bottom-up splay-boom.
- (b) Hoe kunnen de basisbewerkingen *zoek*, *voegToe*, *verwijder* en *verwijderKleinste* geïmplementeerd worden a.h.v. de *splay*-bewerking? Formuleer deze algoritmen in pseudocode.
- (c) Bespreek de tijdscomplexiteit van deze basisbewerkingen. (geen expliciet bewijs gevraagd)
- (d) Vergelijk de bruikbaarheid van splay-bomen met gebalanceerde zoekbomen, zoals AVL-bomen en rood-zwart-bomen.

## 3. Oefening : Het handelsreizigerprobleem (4 pten.)

- (a) Geef een algoritme dat geschikt is voor het oplossen van een handelsreizigerprobleem met 30 steden. Motiveer waarom dit algoritme daarvoor geschikt is; bespreek voordelen en nadelen van dit algoritme.
- (b) Geef een algoritme dat geschikt is voor het oplossen van een handelsreizigerprobleem met 10000 steden. Motiveer waarom dit algoritme daarvoor geschikt is; bespreek voordelen en nadelen van dit algoritme.

## 4. Ontwerpoefeningen (4 pten.)

- (a) Vertaal het volgende probleem naar een gekend probleem waarvoor we in de cursus (zowel theorie- als oefeningengedeelte) een algoritme gezien hebben. Geef enkel aan hoe het probleem kan vertaald worden; het is **niet** nodig ook het algoritme uit te werken.

In een schrijnwerkerij moet een reeks planken van gegeven lengte (niet alle planken hebben dezelfde lengte) gezaagd worden uit stukken hout van een zekere vaste lengte  $x$ . Dit moet gebeuren met zo weinig mogelijk verspilling van hout.

- (b) Geef aan hoe de geziene datastructuren kunnen gebruikt worden om het volgende probleem op te lossen.

Gegeven een collectie data waarbij elke record bestaat uit 2 datavelden, bijvoorbeeld: naam (voor de eenvoud beschouwd als één enkele string) en telefoonnummer (een string). We willen in deze collectie snel kunnen opzoeken, zowel op naam als op telefoonnummer. Bovendien willen we zowel een alfabetische namenlijst als een lijst gesorteerd op telefoonnummer kunnen opvragen.

---