

## ALGORITMIEK: opgaven werkcollege 8

**Partitie (1 t/m 2) Divide and conquer (3 t/m 4);  
Decrease and conquer (5 t/m 7); Decrease by half (8);**

**Opgave 1.** (Levitin, 5.2.8 (zie tweede editie, opgave 4.2.8))

Reorganiseer de elementen van een gegeven array  $A$  zodanig dat alle negatieve waarden voorafgaan aan de positieve. Het algoritme moet lineair zijn (en slechts één doorgang door het array maken) en in situ (alleen interne verwisselingen). Vergelijk het partitioneren (Partitie) van het array bij Quicksort.

**Opgave 2.** (Levitin, 5.2.9.a: (zie tweede editie, opgave 4.2.9) Dutch National Flag Problem)

Gegeven een array gevuld met 'R', 'W', en 'B'. Reorganiseer dit array zo dat van links naar rechts eerst alle 'R', dan de 'W' en dan de 'B' komen te staan. Het algoritme moet lineair zijn en in situ (alleen interne verwisselingen). Het mag maar één doorgang door het array maken.

**Opgave 3.** (Levitin, 5.4.2 (vgl. tweede editie, opgave 4.5.2))

Bereken  $1201 * 2430$  door het op college en Levitin 5.4 (zie tweede editie, paragraaf 4.5) beschreven verdeel & heers (divide & conquer) algoritme toe te passen.

**Opgave 4.** (naar Levitin, 5.5.1 (zie tweede editie, opgave 4.6.1))

**a.** We bekijken het eendimensionale closest-pair probleem: gegeven een verzameling van  $n$  reële getallen, vind de twee getallen die het dichtst bij elkaar liggen. Geef een verdeel & heers algoritme en sorteer de punten eerst.

**b.** Geef ook een niet-recursief algoritme.

**Opgave 5.** (zie tweede editie, opgave 5.4.7)

Schrijf een decrease by one algoritme (in pseudocode of C++) dat alle  $2^n$  bitstrings van lengte  $n$  genereert.

**Opgave 6.**

Lees op bladzijde 173-174 van Levitin (vgl. tweede editie, bladzijde 181) wat Gray-codes zijn en maak vervolgens de volgende opgave.

**a.** (Levitin, 4.3.9.a (zie tweede editie, opgave 5.4.9)) Gebruik de decrease-by-one techniek om de Gray-code voor  $n = 4$  te genereren.

**b.** (Levitin, 4.3.9.b (niet in tweede editie)) Pas het volgende niet-recursieve algoritme om een Gray-code te genereren toe voor  $n = 4$ :

Begin met een string met  $n$  0'en.

Voor  $i = 1, 2, \dots, 2^n - 1$ , genereer de  $i$ -de bitstring door in de vorige bitstring het  $b$ -de bit te 'flippen', waarbij  $b$  de positie is van de minst significante (= achterste) 1 in de binaire representatie van  $i$ .

**-volgende opgave: z.o.z.**

**Opgave 7.** (uit een oud tentamen)

Gegeven een array  $A$  ( $A[1], \dots, A[n]$ , met  $n \geq 2$ ), dat evenveel oneven als even getallen bevat. De oneven getallen staan op de oneven posities en de even getallen op de even posities. Het array moet —via verwisselingen— zo gereorganiseerd worden dat alle oneven getallen vooraan komen te staan, en alle even getallen achteraan.

- a. Geef een eenvoudig iteratief algoritme voor dit probleem dat slechts één for-loop gebruikt. Hoeveel verwisselingen doet je algoritme?
- b. Geef een decrease by four algoritme (in pseudocode of C++) voor dit probleem. Neem hierbij aan dat  $n$  een 2-voud is. Er moet dus een recursieve functie `hussel(i, j)` worden geschreven die het probleem oplost voor het deelarray  $A[i], \dots, A[j]$  ter lengte een 2-voud.

**Opgave 8.** (uit een oud tentamen)

Gegeven een array  $A$  met  $n$  ( $\geq 1$ ) verschillende gehele getallen  $A[1], A[2], \dots, A[n]$ . Verder is gegeven dat er een index  $p$  met  $1 \leq p \leq n$  bestaat zodat  $A$  stijgend is tot index  $p$ , en daarna dalend.

*Voorbeeld:* als  $A = 3 \ 6 \ 9 \ 11 \ 8 \ 2$ , dan is  $p = 4$ . *Randgevallen:* als  $A = 7 \ 5 \ 3 \ 2 \ 1$ , dan  $p = 1$ ; als  $A = 5$  (dus bestaat uit 1 element), dan  $p = 1$ ; als  $A = 4 \ 9$ , dan  $p = 2$ .

Geef een decrease-by-half algoritme (in pseudocode of C++) voor het bepalen van deze index  $p$  en leg uit waarom het werkt.