**Bedrijfskunde**

Leerjaar 2 – periode 8

Adam Oubelkas – IO2B4

*Hoofdstuk 6 - Algoritme*

Opdracht

Stel ik heb nu de getallen 1 t/m 1000. Hoeveel verschillende arrays heb ik nu nodig?

Stappen

1. Beschrijf met woorden hoe jouw algoritme het probleem gaat oplossen. Schrijf het op en bespreek het met de docent.
2. Analyseer of je algoritme misschien wel slimmer/sneller/efficiënter kan.
3. Programmeer je code in je favoriete programmeertaal. Toon het resultaat aan je docent. Hoeveel arrays hebben 1000 getallen nodig?

4) Wat is de oplossing als ik nu 10000 getallen heb? Hoeveel als ik 100000 getallen (van 1 tot 100000) heb?

5) Kan je je algoritme nog verder optimaliseren?

*Wil je arrays slim aanpakken, dan is het handig als ieder array een naam heeft die in de UI weergeven kan worden, zodat de eindgebruiker een duidelijk overzicht krijgt.*

**Als je efficiënt (sneller algoritme) gebruik wilt maken van arrays (lists), doe dan het volgende:**

*Vraag het gewenste getal van de eindgebruiker.*

*Begin altijd vanaf het getal 1 en voeg het volgende getal toe in de array (list).*

*Als het getal groter is dan of gelijk aan 2 (minimaal 2 getallen moeten vooraf in de array zijn), bepaal of de rest, van een willekeurig getal (getal moet nooit 0 zijn) gedeeld door het huidige getal in een array, gelijk is aan 0.*

**Wanneer je zo veel mogelijk arrays wilt maken (optimaal), kun je beide van deze formules toepassen:**

*Vraag het gewenste getal van de eindgebruiker.*

*Begin altijd vanaf het getal 1 en voeg het volgende getal toe in de array (list).*

*Vergelijk de uitkomst van het huidige getal met het eerste getal binnen een array: als het laatste getal een veelvoud is van het eerste getal, moet het huidige getal in een nieuw array.*

*Herhaal het vergelijkingsproces net zo lang totdat het totaalaantal elementen over alle gemaakte arrays gelijk is aan het ingevoerde getal van de eindgebruiker.*

*De formule hiervoor ziet er als volgt uit:*

*Maak nieuwe array als huidige getal in huidige array / eerste getal in huidige array >= 2*

*Voeg tenslotte alle gemaakte arrays samen tot een (datatype) multidimensionale array = new datatype[aantal arrays(), aantal elementen in arrays].*

*Voorbeeld:*

*Stel je wilt 20 getallen (1 t/m 20) verdelen over zo weinig mogelijk arrays. Hoeveel arrays heb je dan nodig?*

*1 / 1 = 1; in huidige array; totaal arrays: 1*

*2 / 1 = 2; in nieuwe array; totaal arrays: 2*

*2 / 2 = 1; in huidige array*

*3 / 2 = 1,5; in huidige array*

*4 / 2 = 2; in nieuwe array; totaal arrays: 3*

*4 / 4 = 1; in huidige array*

*5 / 4 = 1,25; in huidige array*

*6 / 4 = 1,5; in huidige array*

*7 / 4 = 1,75; in huidige array*

*8 / 4 = 2; in nieuwe array; totaal arrays: 4*

*8 /8 = 1; in huidige array*

*9 / 8 = 1,125; in huidige array*

*10 / 8 = 1,25; in huidige array*

*11 / 8 = 1,375; in huidige array*

*12 / 8 = 1,5; in huidige array*

*13 / 8 = 1,625; in huidige array*

*14 / 8 = 1,75; in huidige array*

*15 / 8 = 1,875; in huidige array*

*16 / 8 = 2; in nieuwe array; totaal arrays: 5*

Mijn programma ziet er in een notendop zo uit: vraag invoergetal > vraag voorkeur algoritme en voorkeur naamgeving > bereken gewenste algoritme > resultaat weergave arrays