

DAKI Programmeeropdracht 2: Schijven

(Deze versie is van 21 april 2020, 23:04)

Je moet deze programmeeropdracht **zelf** en **alleen** maken. Je mag zeker met anderen overleggen over je aanpak, maar code van anderen bekijken of overnemen of zelf code delen met anderen is uitdrukkelijk niet toegestaan. Je moet je programma schrijven in C#, en inleveren via [DOMjudge](#).

1 Opdrachtbeschrijving

Na al dat gegoochel met getallen ontdekt DAKIBOT een interne *desire* om eens iets met haar handen te doen. Je ontvangt via via een verzoek om DAKIBOT in te zetten om in een fabriek de steunblokken te produceren die klanten bestellen. Je hebt nog steeds weinig te doen, behalve online wat vakjes te volgen aan de UU, en besluit de tijd die je uitspaart doordat je in de eerste drie periodes heel efficiënt hebt leren studeren te gebruiken om de code schrijven die DAKIBOT hierbij helpt. (De tijd die je uitspaart doordat je niet naar de Uithof hoeft te fietsen besteedt je aan andere fietstochtjes, waarbij je vanaf of richting de Uithof fietst, en bij zwembad de Kromme Rijn richting Rhijnauwen rijdt teneinde je fiets bij het theehuis te parkeren, om (binair?) een paar van die [geinige geocaches](#) te zoeken die daar in de bossen verstopt liggen.)

Elk door een klant bestelde **steunblok** heeft een bepaalde hoogte, die wordt bereikt door op één zo groot mogelijk **standaardblok** een aantal **schijfjes van hoogte 1** te lijmen. Er is een catalogus met standaardblokken beschikbaar, geordend op grootte, waarin je allereerst moet zoeken naar het geschikte blok. Vervolgens bepaal je hoeveel schijfjes daar nog bovenop moeten worden gelijmd. Een testcase bestaat uit de opdrachten van een aantal klanten, en DAKIBOT moet per testcase het *minimale* aantal schijfjes geven, dat nodig is om alle bestelde steunblokken in de testcase te kunnen produceren. Als de catalogus bijvoorbeeld standaardblokken van hoogte 2, 13, 18 en 23 vermeldt, kan DAKIBOT voor een besteld steunblok van hoogte 21 het standaardblok van 23 niet gebruiken, want dat is te groot. Een standaardblok van 13 met daar bovenop 8 schijfjes zou kunnen, maar we kunnen een nog groter standaardblok van 18 kiezen, waarop we namelijk slechts 3 schijfjes hoeven te lijmen.

2 Invoer en Uitvoer

De invoer heeft op de eerste regel een getal $n \leq 9\,999\,999$, het aantal verschillende verkrijgbare standaardblokken. Dan volgen n regels met een getal, elk maximaal 999 999 999, de hoogte van een standaardblok; deze lijst van standaardblokken is op hoogte gesorteerd. De regel daarna bevat een getal $m \leq 9\,999\,999$, het aantal steunblokken dat DAKIBOT moet afhandelen. Dan volgen m regels met elk een getal, maximaal 1 999 999 999, de hoogte van het gewenste

steunblok. Al deze getallen zijn geheel en niet negatief. Na elk getal kan een spatie staan en andere dingen (zoals in het voorbeeld hieronder, dat één van de testcases is), die moet je dan negeren. De uitvoer heeft 1 regel met 1 getal, namelijk het *totaal aantal schijfjes* dat DAKIBOT nodig heeft.

3 Voorbeeld

Een besteld steunblok kleiner dan het kleinste standaardblok bestaat geheel uit schijfjes. Van een standaardblok kun je meerdere exemplaren gebruiken, zoals het standaardblok van 18 in de volgende testcase. Bij deze invoer:

```
3 // Omvang catalogus
13 // Kleinste standaardblok
18 // Elk blok is max 999999999
23 // De catalogus is gesorteerd
4 // Aantal te produceren steunen
18 // Blok van 18 past precies
25 // Blok van 23 plus twee schijfjes
5 // Alleen vijf schijfjes
21 // Gebruik nog een blok van 18, met drie schijfjes
```

hoort de volgende uitvoer:

```
10
```

4 Algoritmische eisen

Dit kan op verschillende manieren, maar je moet *zelf* Binary Search implementeren om per besteld steunblok te bepalen welk standaardblok DAKIBOT kan gebruiken. Deze aanpak moet een rekentijd van $O(m \lg n)$ geven. Het moet ook inderdaad zo snel, want zowel het aantal bestelde steunblokken, als het aantal standaardblokken in een testcase kan nogal oplopen—maar tegelijkertijd kunnen de standaardblokken zelf best klein zijn ...!

5 Hints, Tips

Bestudeer aandachtig (en beantwoord) de vraag hierover in werkcollege 1. Zorg dat je aan de hand van de tekst in die vraag goed nadenkt over hoe dit eruit moet zien, zodat je alleen de pseudocode nog maar hoeft in te vullen. Bedenk ook waarom ik in de laatste zin van sectie 4 zoveel nadruk leg op de hoogte van de standaardblokken.