

DAKI Programmeeropdracht 1: Collatz

(Deze versie is van 7 september 2022, 13:39)

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

Deze opdracht is bedoeld om je kennis te laten maken met het lezen van invoer, het printen van uitvoer, en het gebruik van [het DOMjudge systeem](#). In de [DOMjudge teamhandleiding](#) vind je in appendix A code voorbeelden in een aantal programmeertalen voor een soortgelijk *eenvoudig* probleem. Deze opdracht is met nadruk ook bedoeld om je te leren dat je opdrachtbeschrijvingen goed moet lezen.

Je bent al een heel eind gevorderd met je studie Kunstmatige Intelligentie. Omdat je door omstandigheden al geruime tijd aan huis gebonden bent, en dus zowel het internet als netflix uit hebt, heb je een robot gemaakt, die je “Dankzij Autonomie Kan Ik Bijzonder Overtuigend Telestuderen”, kortweg *DAKIBOT* besluit te noemen. Je grijpt de gelegenheid die deze kalme onderwijsperiode biedt om je ‘vrije’ uurtjes in het programmeren van DAKIBOT te stoppen, met beide handen aan.

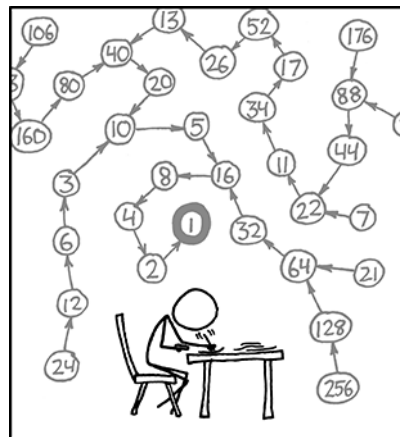
De eerste, nog wat bescheiden, opdracht die je jezelf geeft, is het bevestigen—zo niet bewijzen—van een beroemd wiskundig vermoeden, [het vermoeden van Collatz](#). Neem een positief geheel getal, c . Als het even is, is het volgende getal $\frac{c}{2}$, en anders is het volgende getal $3c + 1$. Blijf deze stap steeds herhalen met het volgende getal dat je vindt. Het vermoeden zegt dat de rij getallen die je aldus produceert, voor elke $c \geq 1$, op 1 uitkomt. Als we met $c = 22$ beginnen, bijvoorbeeld, krijgen we

$22 \rightarrow 11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$.

Deze ‘Collatz-rij’ bevat 15 stappen en de ‘Collatz-lengte’ van de input 22 is daarom 15. Je ziet dat de getallen in de rij makkelijk véél groter dan het begingetal worden... (Wat is bijvoorbeeld het grootste getal in de Collatz-rij van 27?) Je gaat een programma maken dat voor een gegeven getal de Collatz-lengte teruggeeft.

2 Invoer en uitvoer

De eerste regel van de invoer is een getal ≤ 200 , mogelijk gevolgd door een spatie en andere karakters. Dat getal, n , is het aantal getallen waarvoor je programma de Collatz-lengte moet geven. Daarna volgen n regels met op elk van die regels één geheel, positief



THE COLLATZ CONJECTURE STATES THAT IF YOU PICK A NUMBER, AND IF IT'S EVEN DIVIDE IT BY TWO AND IF IT'S ODD MULTIPLY IT BY THREE AND ADD ONE, AND YOU REPEAT THIS PROCEDURE LONG ENOUGH, EVENTUALLY YOUR FRIENDS WILL STOP CALLING TO SEE IF YOU WANT TO HANG OUT.

Figuur 1: Een xkcd over het Collatz vermoeden (<https://xkcd.com/710/>).

getal, $\leq 1\,000\,000$, mogelijkwerwijs gevolgd door een spatie en andere karakters. De uitvoer moet bestaan uit n regels: de Collatz-lengte van elk van de n invoergetallen.

3 Voorbeeld

Bij de volgende invoer:

```
2      % het aantal getallen
15     // het eerste getal
22     !!!!! en het tweede
```

hoort de volgende uitvoer:

```
17
15
```

4 Algoritmische eisen

Geen. Je hebt niets meer nodig dan wat je bij *Modelleren en Programmeren* hebt geleerd. De voor de hand liggende methode om de Collatz-lengte uit te rekenen is geschikt (iets beters is niet bekend), en voor de getallen waar het hier om gaat ($\leq 1\,000\,000$) is bekend dat de Collatz-lengte eindig is en niet 'al te lang'.

5 Hints, Tips

Een algemene tip bij alle opdrachten is dat je goed op de specificatie van de invoer moet letten. Als je vroeg leert **goed te lezen**, voorkomt dat een hoop fouten en frustraties. **TEST BOVENDIEN ALTIJD DE VOORBEELD INPUT in sectie 3 van dit document** in je eigen ontwikkelomgeving (Visual Studio?). De voorbeeld testcase in sectie 3 van de

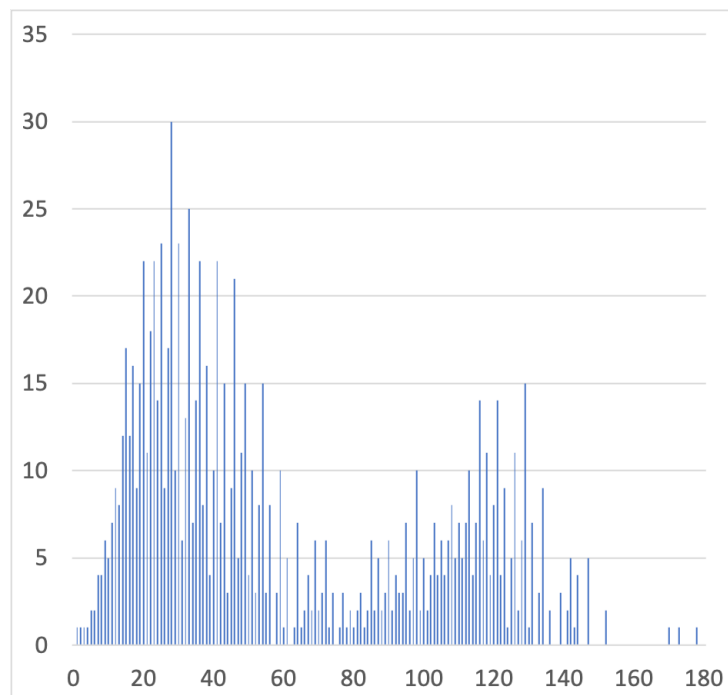
opdrachtbeschrijving is namelijk bij elke programmeeropdracht *letterlijk* de eerste testcase op DOMjudge—inclusief de comments!!! In het bijzonder ga je op DOMjudge foutmeldingen krijgen als je voor de eerste regel de opdracht

```
int aantal = int.Parse(Console.ReadLine());
```

gebruikt! De tekst die op de eerste regel zou kunnen volgen op het getal (in sectie 3 hierboven het getal “2”) is namelijk niet als `int` te parsen! Ik raad je in dit verband aan, te kijken naar de mogelijkheden die de `Split` methode van de `String` klasse biedt.

Je hoeft op een Mac niet in MyWorkPlace te ontwikkelen zoals bij Modelleren en Programmeren, want je ontwikkelt bij DAKI geen programma's met een grafische user interface, maar enkel programma's met een commandline interface, en die kun je prima in Visual Studio voor de Mac bouwen en testen.

Wat Collatz betreft, kun je je programma aanpassen om een aantal interessante vragen te onderzoeken, zoals de volgende. Hoeveel keer zo groot als het begingetal kan een getal in de Collatz-rij worden? Zijn er getallen die gelijk zijn aan de lengte van hun eigen Collatz-rij? Zijn er 'Collatz-vrienden' (twee getallen die gelijk zijn aan de lengte van elkaars Collatz-rij)? Voor welke getallen zitten er meer/evenveel/minder oneven dan even getallen in de Collatz-rij? Je kunt vast zelf ook nog wel leuke vragen bedenken. In figuur 2 zie je bijvoorbeeld hoe vaak elke waarde op de x -as voorkomt als Collatz-lengte van getallen $< 1\,000$. Waarom komt 28 zo vaak voor? En waarom is dit een bimodale verdeling?



Figuur 2: Het aantal keren dat een bepaalde Collatz-lengte voorkomt, voor de getallen $< 1\,000$.