

DomJudge-Practicum

Open Dag UU

1 Introductie

Bij veel vakken die je volgt tijdens je studie informatica aan de UU, moet je programmeeropdrachten maken. Soms moet je die inleveren zodat ze door de docent nagekeken kunnen worden, maar steeds meer vakken gebruiken het automatische nakijksysteem *DomJudge*.

Als je een programma in DomJudge inlevert, wordt het automatisch getest met een aantal testinvoeren. Afhankelijk van hoe je programma werkt, krijg je binnen een paar minuten een beoordeling terug:

- **Correct:** je hebt de opgave goed.
- **Wrong-Answer:** de uitvoer van je programma klopte niet.
- **Timelimit:** je programma was te langzaam.
- **Run-Error:** je programma liep vast met een foutmelding.

Bij een correct heb je de opdracht gehaald, en anders ... mag je het tot de deadline net zo vaak proberen als je wilt!

2 Opgaven

In de bundel zitten vier opgaven, in volgorde van oplopende moeilijkheid. De eerste twee opgaven zijn al opgelost, maar de oplossing is fout! Lever de opgave in in DomJudge, kijk wat de fout is, en probeer die te verbeteren.

- **Collatz:** schrijf een programma om de lengte van een Collatz-rij te berekenen. Deze opgave is geschikt als je nog helemaal geen programmeer-ervaring hebt.
- **Priem:** zoek gegeven een getal x , het eerste getal $y \geq x$ dat een priemgetal is. Als je al een beetje programmeer-ervaring hebt is deze goed te doen.
- **A2B:** tel het aantal A2B-rijtjes. Voor de meer ervaren programmeurs.
- **Feest:** bouw een TomTom voor carpoolen. Deze opgave is extreem moeilijk, en alleen ouderejaars studenten zullen hem kunnen oplossen. Maar het is misschien wel leuk om te zien, wat je allemaal kan leren binnen de opleiding informatica.

3 Instructies

Bestanden. De bestanden die je nodig hebt voor dit practicum staan in `S:\domjudge`.

Visual Studio. We programmeren in de programmeertaal `C#` en doen dit in de editor “Visual Studio”. Open Visual Studio door te dubbelklikken op het bestand `Collatz.sln` (dit is al voor je gedaan). Je krijgt nu de code te zien (die staat in `Collatz.cs` - dit is het bestand dat je in DomJudge moet inleveren, en *niet* het `.sln`-bestand). Druk op `F5` om het programma te testen.

(Voor gevorderde programmeurs) Als je je eigen programma wil maken kun je via `File` → `New` → `Project` → `C# Console Application` een nieuw project maken.

Inleveren. Open <https://domjudge.cs.uu.nl/dj/opendag/team/> en registreer een account met username en password en log vervolgens in. Klik op *select file...* en selecteer het in te leveren bestand (waarschijnlijk `Collatz.cs` of `Priem.cs`¹). Selecteer in de dropdowns het probleem dat je inlevert (Collatz, Priem of A2B) en de taal (de voorbeeld-programma’s zijn in `C#`). Druk op *submit* - binnen enkele seconden krijg je te zien of je programma correct is of niet.

Oplossen. De geleverde voorbeeldoplossingen zijn fout! Probeer de fout te vinden, verbeter hem, en lever opnieuw in. Is het programma nu wel correct?

¹Dit staat in de gelijknamige submap van de `domjudge`-map.

Collatz-Lengte

DomJudge Practicum Opgave 1, Open Dag 18-19 Nov¹

1 Achtergrond

Van een (positief) getal maak je met het volgende recept een *Collatz-rij*²: Als het getal 1 is bereikt, stopt de rij. Als een getal even is, deel door 2; als het oneven is, vermenigvuldig met 3 en tel er 1 bij. De Collatz-rij van 22 bevat 15 stappen:

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

Zou je na 1 verder gaan, dan kom je na 3 stappen (1, 4, 2, 1) weer bij 1. Je programma moet van vier begingetallen de lengte van de Collatz-rij berekenen.

2 Specificatie van Input en output

De invoer bestaat uit vier regels, die elk precies èèn getal kleiner dan 1000000 bevatten. De uitvoer moet ook vier regels hebben met elk een getal, namelijk de lengte van de Collatz-rij van het zoveelste invoer-getal.

3 Voorbeeld

Voorbeeld ColVb1: Houd rekening met minimale invoer (1)!

15	17
871	178
1	0
22	15

4 Aanwijzingen

Er bestaat een Collatz-anagram: van het startgetal 871 bestaat de lengte uit dezelfde cijfers.

Kun je een getal vinden dat *gelijk is* aan de lengte van zijn Collatz-rij? Kun je twee getallen vinden die *elkaars* Collatz-lengte zijn?

¹Versie: 4 november 2016.

²Zie https://nl.wikipedia.org/wiki/Vermoeden_van_Collatz.

Priemgetal

DomJudge Practicum Opgave 2, Open Dag 18-19 Nov¹

1 Achtergrond

Help! Teus de Timmerman is zijn gereedschap kwijt! Help hem zoeken naar het volgende priemgetal. Vind, voor ieder getal x op de invoer, het kleinste priemgetal $\geq x$ vanaf dat getal.

2 Specificatie van Input en output

De invoer bestaat uit positieve getallen, eentje per regel, gevolgd door een 0. Voor elk positief getal moet je zoeken naar het eerstvolgende priemgetal. De uitvoer heeft de lijst met gezochte priemgetallen, en wordt ook afgesloten met 0.

3 Voorbeeld

Voorbeeld `PriemVb1`: Als het invoergetal priem is, is dat ook de uitvoer. Wat zou er gebeuren als je de uitvoer er weer instopt als invoer?

40	41
1350	1361
2500	2503
2503	2503
222222	222247
999999999	1000000007
0	0

4 Aanwijzingen

Je kunt testen of een getal n priem of een product is, door het te delen door alle getallen tussen 2 en $n - 1$. Gaat zo'n deling geheel op (rest 0) dan is n een product. Als geen van de delingen opgaat, is n priem.

Het delen gaat iets sneller als je niet door alle getallen deelt, maar alleen door 2 en alle oneven getallen. Zal het zoeken ook sneller gaan wanneer je alleen oneven waarden voor n probeert?

Als n een deler d heeft, dan is het getal $e = n/d$ ook een deler. En: het is zeker dat d of e kleinergelijk is aan de wortel van n . (Waarom?)

¹Versie: 4 november 2016.

Tellen van A naar B

DomJudge Practicum Opgave 3, Open Dag 18-19 Nov¹

1 Achtergrond

Een (A, B) -rijtje is een rij tekens I (voor Increment) en D (voor Dubbel) zo, dat als je de operaties toepast op A , er B uitkomt. Voorbeeld: DII is een $(5, 12)$ -rijtje. Begin met 5, verdubbel, verhoog, verhoog en je krijgt 12. Er bestaan drie $(5, 12)$ -rijtjes, namelijk IIIIII, DII, ID. Je programma berekent van K paren A en B , het aantal (A, B) -rijtjes.

2 Specificatie Invoer en Uitvoer

De invoer begint met een regel met één getal, $K < 256$. Daarna volgen K regels met elk twee getallen A en B , waar $0 < A \leq B < 10000$. De uitvoer bestaat uit K regels met elk één getal, op de i^{de} regel het aantal rijtjes van het i^{de} paar.

3 Voorbeeld

Voorbeeld **geen**: Check altijd de minimaal toegestane invoer.

```
0
```

Voorbeeld **klein**: Dit kun je nog met de hand checken.

```
2                8
3 14            13
5 24
```

Voorbeeld **twee**: Het aantal loopt snel op; reken met `longs`, niet met `ints`.

```
2                354
10 100          74116423
10 1000
```

4 Aanwijzingen

Het aantal (A, B) -rijtjes is uit te drukken als het aantal rijtjes dat op I eindigt, plus het aantal dat op D eindigt. Het aantal rijtjes dat op I eindigt, is gelijk aan het aantal $(A, B - 1)$ -rijtjes.

¹Versie: 4 november 2016.

Het Eindfeest

DomJudge Practicum Opgave 4, Open Dag 18-19 Nov¹

1 Achtergrond

Om het (successvol) afsluiten van het vak Algoritmie te vieren, is er een feest georganiseerd. Jij beschikt als enige van je vrienden over een auto. Het feest kan pas beginnen als iedereen er is, dus je besluit om te gaan carpoolen. Je kan je vrienden thuis ophalen, maar je vrienden kunnen ook naar een ophaallocatie lopen en daar opgepikt worden of direct naar het feest lopen. Je vrienden zijn te voet $S \times$ langzamer dan jij met je auto. Jouw taak is om voor jezelf en je vrienden een zo efficiënt mogelijke route te berekenen, d.w.z. een route waarbij het feest zo vroeg mogelijk kan beginnen als iedereen tegelijkertijd vertrekt. Je mag aannemen dat er genoeg plaats in je auto is voor al je vrienden.

2 Specificatie Invoer en Uitvoer

De invoer begint met een regel met daarop vijf gehele getallen: $n \leq 10.000$, $m \leq 100.000$, $k \leq 18$, $S \leq 10.000$ en de uitvoermodus (die 1 of 2 kan zijn). Hierbij is n het aantal mogelijke locaties l_1, \dots, l_n en is k het aantal vrienden.

Dan volgen n regels met op de i^e regel een string: de naam van locatie l_i .

Dan volgen op één regel $k+2$ getallen: het eerste is de locatie van jouw huis, het tweede de locatie van het feest en de overige getallen zijn de locaties van je vrienden v_1, \dots, v_k .

Dan volgen m regels met ieder 3 gehele, positieve getallen. Een paar a, b, c stelt voor dat er een ongerichte verbinding is tussen locaties l_a en l_b die met de auto c minuten kost om te overbruggen.

De uitvoer is een getal X , het aantal minuten tot het feest kan beginnen, gerekend vanaf het moment dat de eerste persoon van huis vertrekt. Indien de uitvoermodus 1 is, volgt hierna een routebeschrijving.

De routebeschrijving bestaat uit een lijst namen van locaties, ieder op een aparte regel, in de volgorde dat je ze bezoekt. Als je vriend i ophaalt wordt dit aangegeven met een regel **Vriend** i . Je mag je vrienden alleen laten instappen op een knooppunt, dit kan uit veiligheidsoverwegingen niet halverwege een verbinding. Je vrienden beschikken zelf over een routeplanner, hun route hoeft je dus niet te geven. Als een vriend zelf naar het feest loopt (en dus niet met de auto meerijdt) dan komt deze vriend niet in de uitvoer.

¹Versie: 17 november 2016.

3 Voorbeeld Invoer en Uitvoer

Voorbeeld voorbeeld: Vriend 3 loopt eerst van Afrit 19 naar School, daar pik je hem op.

9 12 3 10 1	48
Thuis	Thuis
Toren	Toren
Stadhuis	Kasteel
Kasteel	Vriend 1
School	School
Molen	Molen
Afrit 18	Afrit 18
Feest	Vriend 2
Afrit 19	Molen
1 8 4 7 9	School
1 2 3	Vriend 3
1 3 4	Molen
1 8 6	Feest
2 4 6	
2 3 2	
3 5 8	
4 5 5	
5 6 2	
5 9 3	
6 7 12	
6 8 4	
7 9 20	

4 Algoritmische Aanwijzingen

Dit is de lastigste opdracht. Laat je inspireren door twee algoritmen/technieken die op het college zijn behandeld. Misschien is het handig om bitstrings²/bit-operaties te gebruiken.

Kun je bewijzen dat het probleem *NP*-moeilijk is, bijvoorbeeld door een reductie van traveling salesman? Is het probleem in *NP*?

²Niet de `BitArray`-klasse, een bitstring is gewoon een `int` of `long`.