

Finale Nederlandse Informatica Olympiade

4 juli 2020



informatica
olympiade

Problemen

- 1 Drinkwater voor Kubistanië
- 2 Polycultuur
- 3 Vervelende Broer
- 4 Veterstrikdiploma
- 5 Wegwerkzaamheden

Aanwijzingen

Over de opgaven

- Er zijn in totaal 5 opgaven. De wedstrijd duurt 4 uur.
- Lees de opgaven altijd goed door. Alleen het éérste uur mag je vragen stellen.
- De opgaven zijn in willekeurige volgorde geven (dus niet op moeilijkheid gesorteerd).
- Voor elke opgave kun je maximaal 100 punten behalen. Je kunt ook nog deelscores krijgen als je alle testcases goed hebt van een subcategorie met specifieke grenzen.
- Je oplossing moet deterministisch zijn. Dit betekent dat je oplossing bij dezelfde invoer altijd dezelfde uitvoer moet geven.
- Je oplossing moet in C of C++ geschreven zijn.
- **Je inzending wordt alleen gecontroleerd als deze de voorbeelden goed oplost!**

Testen

- Voor de eerste testset van een opgave kun je punten verdienen voor elke test.
- Bij alle volgende testsets van een opgave krijg je alleen punten als je alle tests uit de set goed hebt.
- Je krijgt direct de resultaten te zien. De scores die je ziet zijn onder voorbehoud.

Score

- Bij de tweede ronde kon je maximaal 400 punten verdienen. Deze score noemen we e .
- Je kunt met de set van vandaag maximaal 500 punten verdienen. Deze score noemen we t .
- Voor je eindscore s geldt: $s = t + 0,3 \cdot e$.
- Indien twee of meer deelnemers dezelfde eindscore s hebben dan bepaalt het resultaat van de derde ronde de ranking tussen deze deelnemers onderling.

Huishoudelijk

- Het is niet toegestaan om elektronische hulpmiddelen te gebruiken.
- Het is niet toegestaan om contact over de opgaven te hebben met anderen.
- Het is niet toegestaan om informatie van het internet te halen, behalve de programmeerhandleiding van je programmeertaal, zoals C++-reference of `cplusplus.com`. `StackOverflow` is dus *niet* toegestaan.

Tenslotte

Heel erg veel succes gewenst met de opgaven oplossen!

Deze opgaven zijn opgesteld door Ludo met hulp van Eljakim, Koen, Maarten, Raymond, Willem en Wouter.

1 Drinkwater voor Kubistanië

Tijdslimiet: 3,0s

Het eilandenrijk Kubistanië is omringd door de zee en heeft vanwege langdurige droge periodes behoefte aan een groot waterreservoir om de inwoners van genoeg drinkbaar water te kunnen voorzien. Voor de drinkwatervoorziening wordt er gezocht naar een plek om genoeg water te kunnen opslaan, die een zo klein mogelijk grondoppervlak nodig heeft.

Kubistanië heeft zijn naam te danken aan de merkwaardige indeling van de grond. Elk perceel heeft grenzen die perfect in Noord/Zuid of Oost/West richting geplaatst zijn, en elk stukje grond bestaat uit een gebied van $1\text{ m} \times 1\text{ m}$ op een geheel aantal meters boven (of onder) zeeniveau.

Een waterreservoir is een aaneengesloten gebied waar water in gepompt kan worden tot een hoogte waarbij er geen water kan weglopen naar zee. Water stroomt altijd naar een niet hoger gelegen aangrenzend stuk grond indien er vanuit die plek het naar zee kan stromen. Hierbij worden diagonaal verbonden stukken grond niet als “aangrenzend” beschouwd.

Je kunt ervan uitgaan dat de rand van de kaart altijd zee is, wat inhoudt dat die stukjes grond een strikt negatieve hoogte hebben (boven zeeniveau). Stukken grond die onder zeeniveau liggen maar niet verbonden zijn met de zee, worden niet als zee beschouwd omdat het zeewater hier niet kan komen.

Kun jij de minimale oppervlakte berekenen wat er nodig is om een waterreservoir met $V\text{ m}^3$ water te plaatsen op het eiland zonder dat het water weg kan stromen? Als er geen oplossing mogelijk is, geef dan “Onmogelijk” als output.

Invoer

De invoer bestaat uit:

- Eén regel met drie positieve getallen: V, n, m . Er geldt $1 \leq V \leq 10^9$ en $1 \leq n \times m \leq 2 \cdot 10^5$.
- n regels, elk met m gehele getallen die de hoogte van de $n \times m$ cellen beschrijven. De hoogte van een cel h_{ij} voldoet aan $|h_{ij}| \leq 10^9$.

Uitvoer

Geef als uitvoer een getal wat de minimale oppervlakte is die nodig is om een gebied met capaciteit V te krijgen, of als dit niet mogelijk is, geef dan “Onmogelijk”.

Deelscores

Er zijn vier subcategorieën waarvoor je punten kunt halen. Elke categorie heeft de volgende grenzen op n, m en h_{ij} , bovenop de grenzen al aangegeven onder “Invoer”:

- Voor 10 punten: $n \times m \leq 70$.

- Voor 30 punten: $0 \leq h_{ij} < 1000$ voor elk stuk land en $n, m \leq 100$.
- Voor 20 punten: elk stuk land heeft een andere hoogte.
- Voor 40 punten: geen extra beperkingen.

Voorbeeld invoer 1

```

5 6 6
-1 -1 -1 -1 -1 -1
-1 10 10 10 10 -1
-1 10 1 2 10 -1
-1 10 4 3 10 -1
-1 10 10 10 10 -1
-1 -1 -1 -1 -1 -1

```

Voorbeeld uitvoer 1

```

3

```

Voorbeeld invoer 2

```

2 5 7
-1 -2 -3 -4 -3 -2 -1
-1 2 2 2 2 2 -1
-1 2 1 2 1 2 -1
-1 2 2 2 2 2 -1
-1 -1 -1 -1 -1 -1 -1

```

Voorbeeld uitvoer 2

```

Onmogelijk

```

2 Polycultuur

Tijdslimiet: 2,0s

Je bent in het bezit van een lange strook met landbouwgrond en wilt er graag wat gewassen in verbouwen. Er zijn drie soorten gewassen die je wilt gaan zaaien: boontjes, sla en witte kool.

Het akker is rechthoekig en opgedeeld in stukken land van 100 m bij 100 m. Qua indeling van de gewassen over de stukken, wil je een zogenaamde ‘polycultuur’ aanleggen. Dit houdt in dat je meerdere soorten gewassen zaait op het akker, op zo’n manier dat de soorten afwisselend zijn.

In een monocultuur wordt er op een akker maar één soort gezaaid waardoor een ziekte heel makkelijk over het hele akker kan verspreiden als het eenmaal aanwezig is. Bij een polycultuur heeft een ziekte weinig effect omdat het zich niet kan verspreiden naar een naastgelegen stuk met een andere soort.

Je wilt nu een akker aanleggen waarbij de kans op ziekteverspreiding minimaal is. Dat wil zeggen, dat twee naast elkaar gelegen (horizontaal of verticaal) stukken land niet met hetzelfde gewas worden bezaaid.

Uit interesse, wil je nu weten op hoeveel manieren je het akker kunt inzaaien met de drie gewassen zodanig dat de ziekteverspreiding minimaal is. Twee manieren van inzaaien zijn anders als er een stuk land is waar bij de één een ander gewas wordt gezaaid dan bij de ander. Aangezien dit aantal heel groot kan zijn, wil je weten wat de rest is na deling door $10^9 + 7$.

Invoer

De invoer bestaat uit één regel met twee positieve getallen n en m ($1 \leq n \leq 10^{18}$ en $1 \leq m \leq 6$).

Uitvoer

Geef het aantal manier om het akker te vullen met de gewassen, modulo $10^9 + 7$. Hierbij is a modulo b , $a \bmod b$, de rest na deling van a door b wat in C++ met de operator `a % b` gedaan kan worden.

Deelscores

Er zijn drie subcategorieën waarvoor je punten kunt halen. Elke categorie heeft de volgende grenzen op n en m , *bovenop* de grenzen al aangegeven onder “Invoer”:

- Voor 20 punten: $n \times m \leq 25$.
- Voor 70 punten: $n \leq 5\,000$.
- Voor 10 punten: geen extra beperkingen.

Voorbeeld invoer 1

1 3	12
-----	----

Voorbeeld uitvoer 1**Voorbeeld invoer 2**

2 3	54
-----	----

Voorbeeld uitvoer 2**Voorbeeld invoer 3**

5 5	580986
-----	--------

Voorbeeld uitvoer 3

3 Vervelende Broer

Tijdslimiet: 1,0s

David kreeg twee 2-dimensionale tabellen A en B , bestaand uit N rijen en M kolommen, aangeleverd met in elk vakje een kleine letter. Deze twee tabellen waren alleen nog niet helemaal naar de wensen van David omdat ze op sommige plekken van elkaar verschillen en daarom wilde hij dit oplossen. Hij doet dit door een rij of kolom te kiezen in A en de tekst hierin om te keren, door de letters als het ware te spiegelen, bijvoorbeeld ‘omkeren’ wordt ‘nerekmo’.

Echter, heeft David een broertje die hem dwars zit: als David rij i van A omkeert, keert zijn broertje rij $N - i + 1$ van tabel A om. Wanneer David kolom j van tabel A omkeert, keert zijn broertje kolom $M - j + 1$ om in A .

Omdat David genoeg krijgt van zijn broertje, vraagt hij jou om hulp bij het herordenen van de tabel. Vind een lijst met operaties die David moet doen zodat A hetzelfde wordt als B , of zeg dat dit op geen enkele manier mogelijk is.

Invoer

De invoer bestaat uit:

- Eén regel met drie niet-negatieve getallen: N, M en `type` waarbij $N \leq M, 1 \leq N \times M \leq 10^6$ en `type` $\in \{0, 1\}$. Het aantal rijen is N , M is het aantal kolommen en `type` geeft het soort testgeval aan (zie “Uitvoer”).
- N regels volgen, met elk M kleine letters zonder spaties ertussen, die een rij van de tabel A beschrijven.
- Daarna volgen nog N regels die tabel B beschrijven.

Uitvoer

Geef als uitvoer “-1” (zonder de aanhalingstekens), als het onmogelijk is.

Zo niet, geef dan op de eerste regel van de uitvoer, een getal K , waarbij K het aantal operaties is wat er volgt. Op de volgende K regels, geef je als uitvoer de operatie die David moet doen op tabel A , wat bestaat uit een karakter “R” of “K” gevolgd door een getal i . Hierbij geeft “R i ” aan dat je rij i ($1 \leq i \leq N$) in tabel A wilt omkeren, en “K i ” geeft aan dat je kolom i wilt omkeren in tabel A ($1 \leq i \leq M$).

De rijen zijn genummerd 1 tot en met N van boven naar beneden, en de kolommen zijn van links naar rechts genummerd $1, 2, \dots, M$.

Als `type` = 0, is elke K toegestaan die voldoet aan $0 \leq K \leq 10^6$. Als `type` = 1, dan moet K gelijk zijn aan het minimale aantal operaties wat vereist is om A gelijk aan B te krijgen.

Deelscores

Er zijn zeven subcategorieën waarvoor je punten kunt halen. Elke categorie heeft de volgende grenzen op N , M en `type`, *bovenop* de grenzen al aangegeven onder “Invoer”:

1. Voor 8 punten: $N = 2$, $M = 2$ en `type` = 1.
2. Voor 13 punten: $N \times M \leq 10$, `type` = 1.
3. Voor 9 punten: $N = 2$, `type` = 1.
4. Voor 7 punten: $3 \leq N$, $N \times M \leq 100$, `type` = 1.
5. Voor 13 punten: $N \times M \leq 1000$, `type` = 1.
6. Voor 23 punten: `type` = 0.
7. Voor 27 punten: `type` = 1.

Voorbeeld invoer 1

2 2 1 ab ba ba ab	1 K 1
-------------------------------	----------

Voorbeeld uitvoer 1

Voorbeeld invoer 2

2 4 1 cccc ddcd dcdd cccc	3 K 1 K 2 R 1
---------------------------------------	------------------------

Voorbeeld uitvoer 2

4 Veterstrikdiploma

Tijdslimiet: 1,0s

In Fibonaccië wil de koning dat de prinses snel haar veterstrikdiploma zal halen. Daarom heeft hij twee docenten ingehuurd om haar te leren strikken; een docent voor de linkerschoen en een docent voor de rechterschoen.

De docenten krijgen een beloning, die natuurlijk groter dan 0 is, van waarde l respectievelijk r . De docenten mogen hun beloning uitzoeken uit een lange rij van oplopend genummerd diamanten. De waarde van één diamant is steeds de som van de waardes van zijn twee voorgangers. De eerste diamant heeft waarde 1, de tweede diamant heeft waarde 2. Zo lopen de waardes als volgt op:

$$1, 2, 3, 5, 8, 13, \dots$$

Elk docent mocht diamanten aanwijzen zodat de som van de waardes van de aangewezen diamanten exact gelijk is aan zijn beloning.

Beide docenten hebben de gekozen diamanten opgeschreven in een *string* van ‘0’ en ‘1’. Hierbij geeft het i^{e} teken vanaf de **rechterkant** aan of de docent een diamant gekozen (‘1’), of juist niet gekozen (‘0’) heeft. De *string* is altijd zodanig opgeschreven dat het meest linkse teken in de *string* een ‘1’ is.

De hofmeester constateerde dat het moeilijk kon zijn om beide docenten direct na elkaar te belonen omdat een diamant misschien twee keer was gekozen, of omdat de beide keuzes samen zouden zorgen dat er twee direct naast elkaar gelegen diamanten gekozen waren. De koning vindt het namelijk niet mooi als er grote gaten in zijn rij van diamanten ontstaat.

De hofmeester geeft jou door welke diamanten de beide docenten hebben gekozen. Het verzoek aan jou is om een reeks diamanten op te geven waarvan de waarde exact gelijk is aan de waarde van $l + r$, maar waarbij in deze reeks elke diamant maximaal één keer is gekozen en er geen twee diamanten naast elkaar zijn gekozen.

Invoer

Op de invoer staan twee regels met daarop telkens een *string* van maximaal 1 000 000 tekens. Elk teken is ‘0’ of ‘1’. Het meest linkse teken in de *string* is altijd ‘1’. De *strings* geven aan welke diamanten gekozen zijn door de beide docenten.

Uitvoer

Schrijf naar de uitvoer één enkele *string* in hetzelfde formaat als de invoer dat aangeeft welke diamanten je moet kiezen om op dezelfde waarde als de invoer uit te komen, en waarbij geen enkele diamant meer dan één keer is gekozen, en er geen twee gekozen diamanten direct naast elkaar liggen.

Deelscores

Er zijn drie subcategorieën waarvoor je punten kunt halen. Elke categorie heeft de volgende beperkingen:

1. Voor 10 punten: de strings bestaan uit hoogstens 16 tekens.
2. Voor 35 punten: precies één diamant wordt door zowel links als rechts gekozen.
3. Voor 55 punten: geen extra beperkingen.

Voorbeeld invoer 1

10100100 100101

Voorbeeld uitvoer 1

100010101

5 Wegwerkzaamheden

Tijdslimiet: 1,0s

Er zijn wegwerkzaamheden en er is maar één rijbaan beschikbaar. Met stoplichten wordt het verkeer geregeld. Op tijdstip $T = 0$ seconde springt het stoplicht aan de linkerkant op groen. Dat blijft dan voor S seconde op groen staan. Omdat auto's precies D seconde nodig hebben om het hele afgezette stuk af te leggen, springt het stoplicht aan de rechterkant pas op tijdstip $T = S + D$ seconde op groen. Dat blijft dan voor S seconde op groen staan en gaat dan op rood. Op tijdstip $T = 2S + 2D$ seconde gaat het licht aan de linkerkant weer op groen en begint de cykel opnieuw. Tussen twee achtereenvolgende auto's moet minstens 2 seconde zitten en auto's mogen natuurlijk niet door rood rijden.

Een auto die precies op $T = S$ seconde van de linkerkant aan komt rijden, kan nog net door groen rijden. Dit heeft geen invloed op eventuele auto's die aan de rechterkant staan te wachten. De eerste auto van rechts kan nog steeds op tijdstip $T = S + D$ vertrekken. Met andere woorden, er hoeven niet 2 seconde te zitten tussen auto's die in verschillende richtingen rijden.

Het doel van deze opgave is om de totale vertraging in seconden die alle auto's bij elkaar op hebben gelopen door de wegwerkzaamheden te berekenen.

Invoer

- Op de eerste regel staan vier gehele getallen S, D, L, R . Hierbij zijn L en R het aantal rijen van auto's dat van de linkerkant aankomt en het aantal auto's dat van de rechterkant aankomt, waarbij $1 \leq D, L, R, S \leq 100$.
- Dan volgen L regels met op elke regel drie getallen B, I en E . Iedere regel omschrijft een rij auto's die van links komt. De auto's in de rij komen aan bij het stoplicht op tijdstippen $B, B + I, B + 2I$, tot en met E , waarbij $2 \leq I \leq 10^9$ en $0 \leq B \leq E \leq 10^9$ en I deelt $E - B$.
- Dan volgen R regels met op elke regel drie getallen B, I en E . Iedere regel omschrijft een rij auto's die van rechts komt. De auto's in de rij komen aan bij het stoplicht op tijdstippen $B, B + I, B + 2I$, tot en met E , waarbij $2 \leq I \leq 10^9$ en $0 \leq B \leq E \leq 10^9$ en I deelt $E - B$.

Het is gegarandeerd dat er in totaal hoogstens 10^5 auto's zijn en dat er minstens 2 seconde zit tussen elk tweetal aankomende auto's uit dezelfde richting.

Uitvoer

- De som van de vertragingen van alle auto's bij elkaar opgeteld. De vertraging is de tijd dat een auto stil staat voor het stoplicht, of equivalent: het verschil in aankomsttijd aan de andere kant van de weg met de situatie waarin er geen wegwerkzaamheden geweest zouden zijn.

Deelscores

Er zijn twee subcategoriën waarvoor je punten kunt halen. Elke categorie heeft de volgende grenzen, *bovenop* de grenzen al aangegeven onder “Invoer”:

- Voor 20 punten: de rijen met auto's zijn in oplopende aankomsttijd gegeven en overlappen niet onderling voor links en rechts.
- Voor 80 punten: geen extra beperkingen.

Voorbeeld invoer 1

20	10	1	1
0	67	201	
1	2	3	

Voorbeeld uitvoer 1

97
