

beCP 2026

Taak 1.1: De Verdraaide Sloten (rotation)

Auteur: Bruno Ploumhans

Maximale uitvoeringsduur: 1 s Geheugenlimiet: 16 MB

Opmerking: Dit is een interactieve taak. Kijk de speciale instructies om je programma te implementeren, compileren en testen na. Aarzel niet om een begeleider om hulp te vragen.

In de oude ommuurde stad Khiva, Oezbekistan, hebben archeologen een kluis ontdekt met N sloten die op een cirkel zijn gerangschikt, genummerd van 1 tot N rond de ring. Nu vragen ze om je hulp. Hier is de informatie die ze hebben verzameld. Lang geleden kreeg elk slot een geheime waarde toegekend. Deze waarden zijn eenvoudigweg de getallen 1 tot en met N , maar ze zijn gedraaid; dat wil zeggen, de reeks 1, 2, 3, ..., N is op de cirkel geplaatst en vervolgens gedraaid met een onbekende offset k . Dus slot nummer 1 bevat de waarde $k + 1$, slot nummer 2 bevat $k + 2$, enzovoort, waarbij we terugkeren naar 1 na N ...

De kluis opent wanneer je de offset k identificeert waarmee de cirkel werd gedraaid. Helaas staan de lokale bewakers je niet toe om direct de waarde van een slot te inspecteren, aangezien ze beweren dat je het mechanisme zou kunnen beschadigen. In plaats daarvan kun je alleen een bewaker vragen om antwoord te geven op vergelijkingsvragen: je geeft hem twee slotposities i en j , en hij vertelt je welke van de twee de kleinere waarde heeft. De bewakers hebben beperkt geduld, dus met een beperkt aantal vragen moet je ontdekken met hoeveel de cirkel is gedraaid.

Interactieprotocol

Je moet een functie `solve(int N)` implementeren die door de grader zal worden aangeroepen. Je `solve` functie zal het geheel getal N als argument ontvangen. Binnen deze functie kun je de volgende functie gebruiken die door de interactiebibliotheek wordt geleverd:

- **`int ask_guardian(int position1, int position2)`.**
 - Roep deze functie aan met twee posities. De twee posities moeten tussen 1 en N liggen, inclusief, en verschillen van elkaar.
 - De functie retourneert een van de twee gegeven posities: het slot dat de kleinere waarde bevat.
 - Je mag deze functie maximaal **60** keer aanroepen.

Zodra je de offset k hebt bepaald waarmee de getallen op de sloten zijn gedraaid, moet je `solve` functie deze retourneren om je antwoord te geven. Je moet ervoor zorgen dat $0 \leq k < N$.

Algemene limieten

- $2 \leq N \leq 100,000,000$
- $0 \leq k < N$
- Je mag `ask_guardian` maximaal 60 keer aanroepen.

Bijkomende beperkingen

Subtaak	Punten	Beperkingen
A	27	$2 \leq N \leq 60$.
B	24	$2 \leq N \leq 100$.
C	49	Geen bijkomende beperkingen.

Voorbeeldinteractie

Hier is een voorbeeldinteractie voor $N = 5$ en $k = 4$. In dit geval hebben de sloten de volgende waarden:

Positie	Waarde
1	5
2	1
3	2
4	3
5	4

Stel dat we de volgende reeks aanroepen besluiten uit te voeren:

Jouw Aanroep	Waarde	Rapport van de Bewaker
<code>ask_guardian({1, 2})</code>	2	De kleinste van de twee sloten zit op positie 2.
<code>ask_guardian({2, 3})</code>	2	De kleinste van de twee sloten zit op positie 2.
<code>ask_guardian({3, 4})</code>	3	De kleinste van de twee sloten zit op positie 3.
<code>ask_guardian({2, 5})</code>	2	De kleinste van de twee sloten zit op positie 2.

Uit deze interactie leiden we af dat aangezien het slot op positie 2 kleiner is dan alle andere, het een waarde van 1 moet hebben. We kunnen dus afleiden dat de getallen zijn gedraaid met $k = 4$.

Lokale Testen met Voorbeeldgrader

Hoe te Gebruiken

1. Bewaar de `rotation.h` en `grader.cpp` bestanden in dezelfde map als je oplossingsbestand (wat `rotation.cpp` moet heten).
2. Compileer de voorbeeldgrader samen met je oplossing met:

```
g++ -std=gnu++11 -O2 grader.cpp rotation.cpp -o rotation
```

3. Voer het gecompileerde programma uit:

```
./rotation
```

4. De voorbeeldgrader zal je vragen om de waarde van N en k in te voeren (bijv., 5 4). Voer deze waarden in, gescheiden door spaties.
5. Je `solve` functie wordt vervolgens aangeroepen en zal interactie hebben met de implementatie van `ask_guardian()` van de voorbeeldgrader. De voorbeeldgrader zal het resultaat van je inzending uitvoeren.
6. Alternatief kun je het voorbeeldbestand gebruiken:

```
./rotation < sample1.in
```

Belangrijke Opmerkingen:

- De voorbeeldgrader is alleen voor lokaal testen en kan mogelijk niet alle beperkingen van de officiële grader afdwingen (bijv., het exacte querylimiet).
- Je dient alleen je `rotation.cpp` bestand in te dienen bij de CMS.