

beCP 2018

Taak 1.1: Japans raadsel (riddle)

Auteur: Victor Lecomte

Maximale uitvoeringsduur: 2s Geheugenlimiet: 512 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.

Tijdens een expeditie in het hartje van de Japanse bergen ontmoet je een oud vrouwtje, gehurkt naast het pad. Dat oud vrouwtje stelt je een raadsel: in gedachten heeft ze een tabel p van grootte n die de getallen $0, \dots, n - 1$ allemaal exact één keer bevat, en jij met ze raden.

Omwille van taalkundige problemen kan je haar slechts twee soorten vragen stellen. Voor elke vraag geef je haar twee indices i en j , gehele getallen van 0 tot en met $n - 1$. Haar antwoord hangt af van het type vraag:

- Type A: Ze antwoordt met **true** als $p[i] < p[j]$, anders met **false**.
- Type B: Ze antwoordt met **true** als $p[i] \& p[j]$ strikt positief is (waarbij $\&$ de binaire EN operatie is), anders antwoordt ze met **false**.

Ter herhaling, de binaire EN is een operatie gebaseerd op de binaire voorstelling van getallen. Het resultaat van $a \& b$ is een getal dat op elke positie van zijn binaire voorstelling een 1 heeft als en slechts als a en b beide een 1 op die positie hebben, en anders een 0. Bijvoorbeeld: $a = 12$ en $b = 5$ worden voorgesteld door 1100 en 0101 , dus het resultaat van $12 \& 5$ is 0100 , oftewel 4.

Bijgevolg, voor vragen van type B antwoordt het vrouwtje met **true** als en slechts als er een positie is waar de binaire voorstellingen van $p[i]$ en $p[j]$ beide een 1 hebben.

Als je zoveel vragen mocht stellen als je wilt, was het natuurlijk te gemakkelijk, dus laat het oud vrouwtje je maar to om N_{askLt} vragen van type A en N_{askAnd} vragen van type B te stellen. Kan je met deze beperking de tabel raden?

1 Te implementeren functies

C++	<code>vector<int> guess(int n)</code>
Java	<code>int[] guess(int n)</code>

Gegeven de grootte n , roep `askLt(i, j)` en `askAnd(i, j)` op om de tabel p te raden.

return Een tabel van grootte n , de geraden tabel p .

2 Op te roepen functies

C++	<code>bool askLt(int i, int j)</code>
Java	<code>boolean askLt(int i, int j)</code>
	Stel een vraag van type A voor de indices i, j .
return	Het antwoord op je vraag: $p[i] < p[j]$.
C++	<code>bool askAnd(int i, int j)</code>
Java	<code>boolean askAnd(int i, int j)</code>
	Stel een vraag van type B voor de indices i, j .
return	Het antwoord op je vraag: $(p[i] \& p[j]) > 0$.

3 Algemene limieten

- $n = 2^{11} = 2048$ voor al onze testen, maar we raden je aan met kleinere waarden te werken om je programma te debuggen;
- n is een macht van 2;
- de tabel p is in onze testen willekeurig gekozen (maar het blijft dezelfde over submissions heen).

4 Bijkomende beperkingen

Subtaak	Punten	N_{askLt}	N_{askAnd}
A	10	5 000 000	5 000 000
B	10	50 000	5 000 000
C	35	200	5 000 000
D	20	200	600 000
E	15	200	100 000
F	10	200	35 000

Elke subtaak bestaat uit 10 tests. Je krijgt de punten van de subtaak enkel als je programma elk van de tests correct in binnen de beperkingen van de vraag oplost.

5 Voorbeeld interactie

De grootte is $n = 4$, de gezochte tabel is $p = \{2, 3, 1, 0\}$. Jouw functie wordt dus opgeroepen met `guess(4)`. De volgende oproepen naar `askLt(i, j)` en

`askAnd(i, j)` worden uitgevoerd:

Oproep	Resultaat	Commentaar
<code>askLt(0,1)</code>	true	$2 < 3$
<code>askAnd(0,1)</code>	true	$2 \& 3 = 2$
<code>askAnd(2,3)</code>	false	$1 \& 0 = 0$
<code>askAnd(2,0)</code>	false	$1 \& 2 = 0$
<code>askLt(3,3)</code>	false	$0 \geq 0$
<code>askAnd(1,1)</code>	true	$3 \& 3 = 3$
<code>askAnd(3,1)</code>	false	$0 \& 3 = 0$
<code>askAnd(1,2)</code>	true	$3 \& 1 = 1$
<code>askAnd(0,3)</code>	false	$2 \& 0 = 0$
<code>askLt(2,0)</code>	true	$1 < 2$
<code>askLt(0,2)</code>	false	$2 \geq 1$

De verkregen informatie is voldoende om af te leiden dat de tabel $\{2,3,1,0\}$ is, dus is dat het resultaat van jouw functie. Natuurlijk is dit slechts een voorbeeld en ben je vrij de vragen te gebruiken zoals je zelf wil.

6 Implementatie, compileren en testen

Je moet de functie `guess(n)` implementeren, die in het bestand `riddle.cpp/java` te vinden is. Om te compileren en vervolgens uit te voeren, gebruik je de volgende commando's:

C++	<code>g++ -std=c++11 -Wall grader.cpp riddle.cpp ./a.out < riddle.in</code>
Java	<code>javac grader.java java grader < riddle.in</code>

Om je code om meerdere gevallen te testen, kan je het bestand `riddle.in` aanpassen. De eerste lijn van dat bestand bevat n , en de tweede lijn kan ofwel de inhoud van de tabel p bevatten, ofwel leeg zijn, dan wordt p willekeurig bepaald.

riddle.in (gespecificeerd)	riddle.in (willekeurig)
<pre>4 2 3 1 0</pre>	<pre>16</pre>

7 Indienen en verdict

- Je moet enkel het bestand `riddle.cpp/java` indienen.
- Het verdict “Wrong Answer” kan betekenen dat je te veel vragen van een type gebruikt hebt, of dat het resultaat dat je teruggeeft incorrect is.
- Print niets naar `stdout` in je programma: gebruik nooit `cout`, `printf()` of `System.out.println()`.