



Gefangenen-Challenge

In einem Gefängnis sind 500 Gefangene. Eines Tages bietet die Gefängniswärterin ihnen eine Chance, freizukommen. Sie legt zwei Beutel mit Geld in einen Raum, Beutel A und Beutel B. Jeder Beutel enthält zwischen 1 und N Münzen (inklusive). Die Anzahl an Münzen in Beutel A ist **verschieden** von der Anzahl an Münzen in Beutel B. Die Gefängniswärterin fordert die Gefangenen heraus, den Beutel mit weniger Münzen zu bestimmen.

Neben den Beuteln mit Münzen enthält der Raum auch ein Whiteboard. Zu jedem Zeitpunkt muss eine einzige Zahl auf dem Whiteboard stehen. Anfangs ist die Zahl auf dem Whiteboard 0.

Die Gefängniswärterin bittet die Gefangenen nacheinander in den Raum. Der Gefangene, der den Raum betritt, weiß nicht, welche oder wie viele andere Gefangene bereits im Raum waren. Jedes Mal, wenn ein Gefangener den Raum betritt, liest er die Zahl, die gerade auf dem Whiteboard steht. Nachdem er die Zahl gelesen hat, muss er sich entweder für Beutel A oder für Beutel B entscheiden. Daraufhin **sieht** der Gefangene in den gewählten Beutel **hinein**, sodass er erfährt, wie viele Münzen im Beutel sind. Nun muss der Gefangene eine der folgenden beiden **Aktionen** ausführen:

- Die Zahl auf dem Whiteboard mit einer nichtnegativen ganzen Zahl überschreiben und den Raum verlassen. Beachte, dass er die Zahl ändern oder beibehalten kann. Danach geht die Challenge weiter (bis alle 500 Gefangenen den Raum betreten haben).
- Einen der Beutel als den mit weniger Münzen bestimmen. Damit endet die Challenge umgehend.

Die Gefängniswärterin wird keinen Gefangenen, der das Zimmer verlassen hat, bitten, es noch einmal zu betreten.

Die Gefangenen gewinnen die Challenge, wenn einer von ihnen erfolgreich den Beutel mit weniger Münzen bestimmt. Sie verlieren, wenn einer von ihnen den Beutel falsch bestimmt oder alle 500 von ihnen im Raum waren und nicht versucht haben, den Beutel zu bestimmen.

Bevor die Challenge beginnt, treffen sich die Gefangenen in der Gefängnishalle und entscheiden sich in drei Stufen für eine gemeinsame **Strategie**.

- Sie wählen eine nichtnegative ganze Zahl x , die größte Zahl, die sie je auf das Whiteboard schreiben wollen könnten.
- Für jede mögliche Zahl i auf dem Whiteboard ($0 \leq i \leq x$) legen sie fest, in welchen Beutel ein Gefangener, der beim Betreten des Raums die Zahl i auf dem Whiteboard liest, sehen

soll.

- Sie legen fest, welche Aktion ein Gefangener ausführen soll, nachdem er die Anzahl an Münzen im gewählten Beutel erfahren hat. Das heißt für jede Zahl i auf dem Whiteboard ($0 \leq i \leq x$) und jede Anzahl an Münzen j in dem Beutel, in den er gesehen hat ($1 \leq j \leq N$), ist entweder festgelegt,
 - welche Zahl zwischen 0 und x (inklusive) auf dem Whiteboard stehen soll, oder
 - welchen Beutel er als denjenigen mit weniger Münzen bestimmen soll.

Wenn sie die Challenge gewonnen haben, wird die Gefängniswärterin die Gefangenen nach x weiteren Tagen entlassen.

Deine Aufgabe ist es, eine Strategie für die Gefangenen zu entwickeln, die sicherstellt, dass sie die Challenge gewinnen (unabhängig von der Anzahl der Münzen in Beutel A und Beutel B). Die Punktzahl deiner Einsendung hängt vom Wert von x ab (siehe Abschnitt Teilaufgaben für Details).

Implementierungsdetails

Du sollst die folgende Funktion implementieren:

```
int[][] devise_strategy(int N)
```

- N : die größtmögliche Anzahl an Münzen in jedem Beutel.
- Diese Funktion soll ein Array s von Arrays von $N + 1$ Zahlen zurückgeben, das deine Strategie darstellt. Der Wert von x ist die Länge des Arrays s minus 1. Für jedes i mit $0 \leq i \leq x$ stellt das Array $s[i]$ dar, was ein Gefangener tun soll, wenn er beim Betreten des Raums die Zahl i auf dem Whiteboard liest:
 1. Der Wert von $s[i][0]$ ist 0, wenn der Gefangene in Beutel A sehen soll, oder 1, wenn er in Beutel B sehen soll.
 2. Sei j die Anzahl der Münzen, die er im gewählten Beutel gesehen hat. Der Gefangene soll dann die folgende Aktion ausführen:
 - Wenn der Wert von $s[i][j]$ gleich -1 ist, soll er A als den Beutel mit weniger Münzen bestimmen.
 - Wenn der Wert von $s[i][j]$ gleich -2 ist, soll er B als den Beutel mit weniger Münzen bestimmen.
 - Wenn der Wert von $s[i][j]$ eine nichtnegative ganze Zahl ist, soll er diese Zahl auf das Whiteboard schreiben. Beachte, dass $s[i][j]$ höchstens x sein darf.
- Diese Funktion wird genau einmal aufgerufen.

Beispiel

Wir betrachten den folgenden Aufruf:

```
devise_strategy(3)
```

Sei v die Zahl, die der Gefangene auf dem Whiteboard liest, wenn er den Raum betritt. Eine der korrekten Strategien ist folgende:

- Wenn $v = 0$ (einschließlich der initialen Zahl), sieh in Beutel A.
 - Wenn er 1 Münze enthält, bestimme Beutel A als den mit weniger Münzen.
 - Wenn er 3 Münzen enthält, bestimme Beutel B als den mit weniger Münzen.
 - Wenn er 2 Münzen enthält, schreibe 1 auf das Whiteboard (dabei wird die 0 überschrieben).
- Wenn $v = 1$, sieh in Beutel B.
 - Wenn er 1 Münze enthält, bestimme Beutel B als den mit weniger Münzen.
 - Wenn er 3 Münzen enthält, bestimme Beutel A als den mit weniger Münzen.
 - Wenn er 2 Münzen enthält, schreibe 0 auf das Whiteboard (dabei wird die 1 überschrieben). Beachte, dass dieser Fall nie eintritt, weil sich schließen lässt, dass dann beide Beutel 2 Münzen enthalten müssten, was nicht erlaubt ist.

Um diese Strategie zu wählen, muss die Funktion `[[0, -1, 1, -2], [1, -2, 0, -1]]` zurückgeben. Die Länge des Arrays ist 2, also ist der Wert von x hierfür $2 - 1 = 1$.

Beschränkungen

- $2 \leq N \leq 5000$

Teilaufgaben

1. (5 Punkte) $N \leq 500$ und der Wert von x darf nicht größer als 500 sein.
2. (5 Punkte) $N \leq 500$ und der Wert von x darf nicht größer als 70 sein.
3. (90 Punkte) Der Wert von x darf nicht mehr als 60 sein.

Wenn die Funktion `devise_strategy` in einem der Testfälle ein Array zurückgibt, das keine korrekte Strategie ist, ist deine Punktzahl für diese Teilaufgabe 0.

In Teilaufgabe 3 können Teilpunkte erzielt werden. Sei m der größte Wert von x für die zurückgegebenen Arrays unter allen Testfällen in dieser Teilaufgabe. Deine Punktzahl für diese Teilaufgabe wird nach der folgenden Tabelle berechnet:

Bedingung	Punkte
$40 \leq m \leq 60$	20
$26 \leq m \leq 39$	$25 + 1.5 \times (40 - m)$
$m = 25$	50
$m = 24$	55
$m = 23$	62
$m = 22$	70
$m = 21$	80
$m \leq 20$	90

Sample-Grader

Der Sample-Grader liest die Eingabe im folgenden Format:

- Zeile 1: N
- Zeile $2 + k$ ($0 \leq k$): $A[k]$ $B[k]$
- letzte Zeile: -1

Jede Zeile außer der ersten und der letzten ist ein Szenario. Das Szenario in Zeile $2 + k$ nennen wir Szenario k . In Szenario k sind $A[k]$ Münzen in Beutel A und $B[k]$ Münzen in Beutel B.

Der Sample-Grader ruft zuerst `devise_strategy(N)` auf. Der Wert von x ist die Länge des zurückgegebenen Arrays minus eins. Wenn der Sample-Grader erkennt, dass das zurückgegebene Array nicht den Spezifikationen im Abschnitt Implementierungsdetails entspricht, gibt er eine der folgenden Fehlermeldungen aus und beendet sich:

- `s is an empty array`: s ist ein leeres Array (was keine gültige Strategie ist).
- `s[i] contains incorrect length`: Es gibt einen Index i ($0 \leq i \leq x$), für den die Länge von $s[i]$ nicht $N + 1$ ist.
- `First element of s[i] is non-binary`: Es gibt einen Index i ($0 \leq i \leq x$), für den $s[i][0]$ weder 0 noch 1 ist.
- `s[i][j] contains incorrect value`: Es gibt Indizes i, j ($0 \leq i \leq x, 1 \leq j \leq N$), für die $s[i][j]$ nicht zwischen -2 und x ist.

Sonst produziert der Sample-Grader folgende zwei Ausgaben.

Zuerst wird der Sample-Grader die Ausgabe deiner Strategie im folgenden Format ausgeben:

- Zeile $1 + k$ ($0 \leq k$): Die Ausgabe deiner Strategie für Szenario k . Wenn die Strategie dazu führt, dass ein Gefangener Beutel A als den mit weniger Münzen bestimmt, dann ist die Ausgabe das Zeichen A. Wenn die Strategie dazu führt, dass ein Gefangener Beutel B als den

mit weniger Münzen bestimmt, dann ist die Ausgabe das Zeichen B. Wenn die Strategie dazu führt, dass kein Gefangener einen Beutel als den mit weniger Münzen bestimmt, dann ist die Ausgabe das Zeichen X.

Außerdem schreibt der Sample-Grader die Datei `log.txt` mit dem folgenden Format in das aktuelle Verzeichnis:

- Zeile $1 + k$ ($0 \leq k$): $w[k][0]$ $w[k][1]$...

Die Folge in Zeile $1 + k$ entspricht Szenario k und beschreibt die Zahlen, die auf das Whiteboard geschrieben wurden. Das heißt, $w[k][l]$ ist die Zahl, die der Gefangene, der als $(l + 1)$ -ter den Raum betreten hat, geschrieben hat.