



Il dilemma dei prigionieri

A 500 prigionieri viene proposta una sfida. Il loro guardiano piazza due sacchi di monete in una stanza, il sacco A e il sacco B. Ciascun sacco contiene tra 1 e N monete (inclusi), e il numero di monete in A è **diverso** da quello in B. I prigionieri devono identificare il sacco con meno monete.

Oltre ai sacchi, la stanza contiene una lavagna, su cui all'inizio è scritto il numero 0. Il guardiano fa scegliere ai prigionieri un intero non-negativo x . La lavagna dovrà sempre contenere un solo intero compreso tra 0 e x , inclusi. Se i prigionieri vincono la sfida, verranno liberati dopo x giorni.

Il guardiano farà entrare i prigionieri nella stanza, uno alla volta e in un ordine a sua scelta. Nessun prigioniero verrà fatto entrare più di una volta. Il prigioniero che entra nella stanza non sa quali o quanti altri prigionieri sono entrati prima. Ogni volta che un prigioniero entra, questi:

1. legge il numero scritto sulla lavagna; poi
2. sceglie uno tra i sacchi A e B e lo **ispeziona**, scoprendo quante monete contiene; poi
3. effettua una singola **azione** ed esce.

Le azioni possibili sono:

- Rimpiazzare il numero sulla lavagna con un intero non-negativo a piacere (anche uguale a quello corrente).
- Identificare uno dei due sacchi come quello con meno monete.

I prigionieri vincono la sfida appena uno di loro identifica correttamente il sacco con meno monete. Perdono se uno di essi identifica il sacco sbagliato, o se tutti e 500 sono entrati nella stanza e nessuno ha tentato una identificazione.

Il tuo compito è pianificare una **strategia** deterministica per i prigionieri. In particolare:

- Devi scegliere l'intero x (il massimo numero che può essere scritto sulla lavagna).
- Devi decidere, per ogni numero $0 \leq i \leq x$, quale sacco deve essere ispezionato da un prigioniero che legge i sulla lavagna.
- Devi decidere quale azione deve effettuare un prigioniero dopo l'ispezione: per ogni $0 \leq i \leq x$ e per ogni numero di monete j nel sacco ispezionato ($1 \leq j \leq N$), devi scegliere
 - un numero tra 0 e x (inclusi) da scrivere sulla lavagna, oppure
 - un sacco da identificare.

La strategia deve far vincere i prigionieri indipendentemente dal numero di monete nei due sacchi. Il punteggio della tua soluzione dipenderà dal valore di x scelto (vedi la sezione Subtask).

Dettagli di implementazione

Devi implementare la seguente funzione:

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

- N : il massimo numero di monete in ogni sacco.
- La funzione deve restituire un array s di array di $N + 1$ interi, che rappresenta la tua strategia. Il valore di x è la dimensione di s meno uno. Per ogni $0 \leq i \leq x$, l'array $s[i]$ descrive come deve comportarsi un prigioniero dopo aver letto il numero i sulla lavagna:
 1. Il valore di $s[i][0]$ è 0 se il prigioniero deve ispezionare il sacco A, e 1 altrimenti.
 2. Sia j il numero di monete nel sacco ispezionato. Il prigioniero deve eseguire un'azione a seconda del valore di $s[i][j]$.
 - $s[i][j] = -1$: indentificare il sacco A.
 - $s[i][j] = -2$: indentificare il sacco B.
 - $s[i][j] \geq 0$: scrivere $s[i][j]$ sulla lavagna. Nota che $s[i][j]$ deve essere al più x .
- Questa funzione è chiamata esattamente una volta.

Caso di esempio

Considera la seguente chiamata:

```
devise_strategy(3)
```

Sia v il numero che il prigioniero legge sulla lavagna. Una possibile strategia corretta è la seguente:

- Se $v = 0$, ispeziona il sacco A. Sia a il numero di monete che contiene.
 - $a = 1$: identifica il sacco A.
 - $a = 3$: identifica il sacco B.
 - $a = 2$: scrivi 1 sulla lavagna.
- Se $v = 1$, ispeziona il sacco B. Sia b il numero di monete che contiene.
 - $b = 1$: identifica il sacco B.
 - $b = 3$: identifica il sacco A.
 - $b = 2$: scrivi 0 sulla lavagna. Nota che questo caso non può verificarsi, perché implica che entrambi i sacchi contengono 2 monete.

Per riportare questa strategia, la funzione deve restituire $[[0, -1, 1, -2], [1, -2, 0, -1]]$. La lunghezza dell'array restituito è 2, quindi il valore di x è $2 - 1 = 1$.

Assunzioni

- $2 \leq N \leq 5000$.

Subtask

1. (5 punti) $N \leq 500$, e il valore di x non deve superare 500.
2. (5 punti) $N \leq 500$, e il valore di x non deve superare 70.
3. (90 punti) Il valore di x non deve superare 60.

Se, in un qualunque caso di test, l'array restituito da `devise_strategy` non rappresenta una strategia corretta, il tuo punteggio per il relativo subtask è 0.

Nel subtask 3 puoi ottenere punteggi parziali. Sia m il massimo valore di x tra tutti i casi di test di questo subtask. Il tuo punteggio è calcolato secondo la seguente tabella:

Condizione	Punti
$40 \leq m \leq 60$	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

Grader di esempio

Il grader di esempio legge l'input secondo il seguente formato:

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

Ogni riga eccetto la prima e l'ultima rappresenta uno scenario. Diciamo che lo scenario k è quello descritto nella riga $2 + k$, in cui i sacchi contengono, rispettivamente, $A[k]$ e $B[k]$ monete.

Per prima cosa, il grader di esempio chiama `devise_strategy(N)`. Il valore di x è la lunghezza dell'array restituito meno uno. Se la strategia restituita da `devise_strategy` non è corretta, il grader di esempio stampa uno dei seguenti messaggi prima di uscire:

- `s is an empty array`: s è un array vuoto.
- `s[i] contains incorrect length`: la lunghezza di $s[i]$ non è sempre $N + 1$.
- `First element of s[i] is non-binary`: il valore di $s[i][0]$ non è sempre 0 o 1.

- $s[i][j]$ contains incorrect value: per qualche i, j ($0 \leq i \leq x, 1 \leq j \leq N$), il valore di $s[i][j]$ non è compreso tra -2 e x .

Altrimenti, vengono prodotti due output.

Innanzitutto il grader stampa i risultati prodotti dalla tua strategia secondo il seguente formato:

- riga $1 + k$ ($0 \leq k$): il risultato per lo scenario k
 - Se la strategia identifica il sacco A, il risultato è il carattere A.
 - Se la strategia identifica il sacco B, il risultato è il carattere B.
 - Se la strategia non conduce all'identificazione di un sacco, il risultato è il carattere X.

Inoltre, il grader scrive su un file `log.txt` nella cartella corrente, secondo il seguente formato:

- riga $1 + k$ ($0 \leq k$): $w[k][0] w[k][1] \dots$

La lista di numeri nella riga $1 + k$ corrisponde allo scenario k e descrive i numeri scritti sulla lavagna dai prigionieri, nell'ordine in cui vengono scritti.