



Челендж у в'язниці

У в'язниці є 500 ув'язнених. Одного разу наглядач пропонує їм шанс звільнитися. Він кладе в кімнату два мішки з грошима, мішок А і мішок В. Кожен мішок містить від 1 до N монет включно. Кожен мішок містить **різну** кількість монет. Мета ув'язнених - визначити мішок з меншою кількістю монет.

У кімнаті, крім мішків з монетами, є ще дошка. У будь-який час на дошці має бути написано одне число. Спочатку число на дошці 0.

Потім наглядач просить ув'язнених по одному заходити в кімнату. Ув'язнений, який входить до кімнати, не знає, хто або скільки інших ув'язнених увійшли до кімнати перед ним. Щоразу, коли ув'язнений заходить до кімнати, він читає число, яке зараз записане на дошці. Прочитавши число, він повинен вибрати мішок А або В. Потім ув'язнений **оглядає** вибраний мішок, таким чином, дізнаючись кількість монет у ньому. Потім ув'язнений повинен виконати одну з наступних двох дій:

- Замінити число на дошці невід'ємним цілим числом і вийти з кімнати. Зверніть увагу, що вони можуть або змінити, або залишити поточне число. Після цього челендж продовжується (якщо всі 500 ув'язнених ще не увійшли в кімнату).
- Визначте один мішок як той, у якому менше монет. Це відразу завершує челендж.

Наглядач не буде просити ув'язненого, який вийшов з кімнати, увійти в кімнату знову.

Ув'язнені виграють челендж, якщо один з них правильно ідентифікує мішок з меншою кількістю монет. Вони програють, якщо хтось із них неправильно ідентифікує мішок, або всі 500 з них увійшли в кімнату і не спробували ідентифікувати мішок з меншою кількістю монет.

Перед початком челенджу ув'язнені збираються в тюремній залі та вибирають **стратегію** для змагання в три кроки.

- Вони вибирають невід'ємне ціле число x , яке є найбільшим числом, яке вони можуть написати на дошці.
- Вони вирішують для будь-якого числа i написаного на дошці ($0 \leq i \leq x$), який мішок має оглянути ув'язнений, який читає число i з дошки при вході в кімнату.
- Вони вирішують, яку дію має виконати в'язень у кімнаті, дізнавшись кількість монет у вибраному мішку. Зокрема, для будь-якого числа i написаного на дошці ($0 \leq i \leq x$) і будь-якої кількості монет j побачених у перевіреному мішку ($1 \leq j \leq N$), вони вирішують або
 - яке число від 0 до x (включно) має бути написано на дошці, або

- який мішок слід визначити як такий, що містить менше монет.

Після перемоги у челенджі наглядач звільнить ув'язнених, але лише через x днів.

Ваше завдання полягає в тому, щоб розробити стратегію для ув'язнених, яка гарантувала б їм перемогу у челенджі (незалежно від кількості монет у мішках А та В). Оцінка вашого розв'язку залежить від значення x (докладніше дивіться у розділі Підзадачі).

Деталі реалізації

Ви повинні реалізувати таку процедуру:

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

- N : максимально можлива кількість монет у кожному мішку.
- Ця процедура має повернути масив s масивів $N + 1$ цілих чисел, що представляє вашу стратегію. Значення x - це довжина масиву s мінус одиниця. Для кожного i такого, що $0 \leq i \leq x$, масив $s[i]$ представляє те, що має зробити в'язень, якщо він прочитає число i з дошки, увійшовши до кімнати:
 1. Значення $s[i][0]$ рівне 0, якщо ув'язнений має перевірити мішок А, або 1 якщо в'язень має перевірити мішок В.
 2. Нехай j це кількість монет, побачених у вибраному мішку. Потім ув'язнений повинен виконати такі дії:
 - Якщо значення $s[i][j]$ дорівнює -1 , ув'язнений повинен ідентифікувати мішок А як такий, що містить менше монет.
 - Якщо значення $s[i][j]$ дорівнює -2 , ув'язнений повинен ідентифікувати мішок В як такий, що містить менше монет.
 - Якщо значення $s[i][j]$ є невід'ємним числом, ув'язнений повинен написати це число на дошці. Зауважте, що $s[i][j]$ має бути не більше x .
- Ця процедура викликається рівно один раз.

Приклади

Розглянемо такий виклик:

```
devise_strategy(3)
```

Нехай v позначає число, яке ув'язнений читає з дошки, заходячи в кімнату. Однією з правильних стратегій є наступна:

- Якщо $v = 0$ (включаючи початкове число), перевірте мішок А.
 - Якщо в ньому 1 монета, то визначте мішок А як той, що містить менше монет.
 - Якщо в ньому 3 монети, то визначте мішок В як той, що містить менше монет.
 - Якщо в ньому 2 монети, то напишіть 1 на дошці (замінивши 0).

- Якщо $v = 1$, огляньте мішок В.
 - Якщо в ньому 1 монета, то визначте мішок В як той, що містить менше монет.
 - Якщо в ньому 3 монети, то визначте мішок А як той, що містить менше монет.
 - Якщо в ньому 2 монети, то напишіть 0 на дошці (замінивши 1). Зауважте, що цей випадок ніколи не трапиться, оскільки ми можемо зробити висновок, що обидва мішки містять 2 монети, що неприпустимо.

Щоб повідомити про цю стратегію, процедура має повернути $[[0, -1, 1, -2], [1, -2, 0, -1]]$. Довжина поверненого масиву дорівнює 2, тому для цього поверненого значення, значення x дорівнює $2 - 1 = 1$.

Обмеження

- $2 \leq N \leq 5000$

Підзадачі

1. (5 балів) $N \leq 500$, значення x має бути не більше 500.
2. (5 балів) $N \leq 500$, значення x має бути не більше 70.
3. (90 балів) Значення x має бути не більше 60.

Якщо в будь-якому з тестових випадків масив, повернутий `devise_strategy` не представляє правильну стратегію, оцінка вашого рішення для цієї підзадачі буде 0.

У підзадачі 3 можна отримати частковий бал. Нехай m буде максимальним значенням x для повернених масивів для всіх тестів у цій підзадачі. Ваш бал за цю підзадачу обчислюється відповідно до наступної таблиці:

| Умова | Бали |
|---------------------|----------------------------|
| $40 \leq m \leq 60$ | 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 |

Приклад градера

Градер зчитує вхідні дані в такому форматі:

- 1-й рядок: N
- $(2 + k)$ -й рядок $(0 \leq k)$: $A[k] B[k]$
- останній рядок: -1

Кожен рядок, крім першого та останнього, представляє сценарій. Сценарій, описаний у рядку $2 + k$, будемо називати k -м сценарієм. У k -му сценарії мішок А містить $A[k]$ монет, а мішок В містить $B[k]$ монет.

Градер спочатку викликає `devise_strategy(N)`. Значення x - це довжина масиву, повернутого викликом, мінус одиниця. Тоді, якщо приклад градера виявляє, що масив, повернутий `devise_strategy` не відповідає обмеженням, описаним у Деталях реалізації, він виводить одне з таких повідомлень про помилку та завершує роботу:

- `s is an empty array`: s порожній масив (який не є правильною стратегією).
- `s[i] contains incorrect length`: існує такий індекс i $(0 \leq i \leq x)$, що довжина $s[i]$ не дорівнює $N + 1$.
- `First element of s[i] is non-binary`: існує такий індекс i $(0 \leq i \leq x)$ такий, що $s[i][0]$ не є ні 0, ні 1.
- `s[i][j] contains incorrect value`: існують індекси i, j $(0 \leq i \leq x, 1 \leq j \leq N)$ такі, що $s[i][j]$ не знаходиться між -2 і x .

Інакше градер видає два результати.

Перше, градер виводить результат вашої стратегії в такому форматі:

- $(1 + k)$ -й рядок $(0 \leq k)$: результат вашої стратегії для сценарію k . Якщо застосування стратегії призводить до того, що ув'язнений ідентифікує мішок А як такий, що містить менше монет, тоді результатом буде А. Якщо застосування стратегії призводить до того, що ув'язнений ідентифікує мішок В як такий, що містить менше монет, тоді результатом буде В. Якщо застосування стратегії не призведе до того, що жоден ув'язнений ідентифікує мішок із меншою кількістю монет, тоді результатом буде Х.

Друге, градер записує файл `log.txt` у поточний каталог у такому форматі:

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

Послідовність у рядку $1 + k$ відповідає сценарію k і описує числа, написані на дошці. Зокрема, $w[k][l]$ - це число, яке написав $(l + 1)$ -й в'язень, щоб увійти до кімнати.