



## Desafio do Prisioneiro

Em uma prisão, há 500 prisioneiros. Um dia, o diretor oferece a eles uma chance de se libertarem. Ele coloca dois sacos com dinheiro, o saco A e o saco B, em uma sala. Cada saco contém entre 1 e  $N$  moedas, inclusive. O número de moedas no saco A é **diferente** do número de moedas no saco B. O diretor apresenta um desafio aos prisioneiros. O objetivo dos prisioneiros é identificar o saco com menos moedas.

A sala, além dos sacos de dinheiro, também contém uma lousa. A qualquer momento, apenas um único número deve estar escrito na lousa. Inicialmente, o número na lousa é 0.

Em seguida, o diretor pede aos prisioneiros que entrem na sala, um a um. O prisioneiro que entra na sala não sabe quais ou quantos outros prisioneiros já entraram na sala antes dele. Toda vez que um prisioneiro entra na sala, ele lê o número atualmente escrito na lousa. Depois de ler o número, ele deve escolher entre o saco A ou o saco B. O prisioneiro então **inspeciona** o saco escolhido, descobrindo assim o número de moedas dentro dele. Em seguida, o prisioneiro deve realizar uma das duas seguintes **ações**:

- Sobrescrever o número na lousa com um inteiro não-negativo e sair da sala. Note que ele pode mudar ou manter o número atual. O desafio continua depois disso (a menos que todos os 500 prisioneiros já tenham entrado na sala).
- Identificar um saco como aquele que tem menos moedas. Isto encerra imediatamente o desafio.

O diretor não pedirá a um prisioneiro que tenha saído da sala para entrar novamente na sala.

Os prisioneiros vencem o desafio se um deles identificar corretamente o saco com menos moedas. Eles perdem se algum deles identificar o saco incorretamente ou se todos os 500 prisioneiros tiverem entrado na sala e não tiverem tentado identificar o saco com menos moedas.

Antes de começar o desafio, os prisioneiros se reúnem no corredor da prisão e definem uma **estratégia** em comum para o desafio em três etapas.

- Eles escolhem um inteiro não-negativo  $x$ , que é o maior número que eles podem querer escrever na lousa.
- Eles decidem, para qualquer número  $i$  escrito na lousa ( $0 \leq i \leq x$ ), qual saco deve ser inspecionado por um prisioneiro que lê o número  $i$  na lousa ao entrar na sala.
- Eles decidem que ação um prisioneiro na sala deve realizar após descobrir o número de moedas no saco escolhido. Especificamente, para qualquer número  $i$  escrito na lousa (

$0 \leq i \leq x$ ) e qualquer número de moedas  $j$  visto no saco inspecionado ( $1 \leq j \leq N$ ), eles decidem:

- qual número entre 0 e  $x$  (inclusive) deve ser escrito na lousa **ou**
- qual saco deve ser identificado como aquele que tem menos moedas.

Ao vencer o desafio, o diretor libertará os prisioneiros depois que cumprirem mais  $x$  dias de pena na prisão.

Sua tarefa é elaborar uma estratégia para os prisioneiros que garanta que eles vençam o desafio (independentemente do número de moedas no saco A e no saco B). A pontuação da sua solução depende do valor de  $x$  (veja a seção Subtarefas para detalhes).

## Detalhes de Implementação

Você deve implementar o seguinte procedimento:

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

- $N$ : o número máximo possível de moedas em cada saco.
- Este procedimento deve retornar um vetor  $s$  de vetores de  $N + 1$  inteiros, representando sua estratégia. O valor de  $x$  é o tamanho do vetor  $s$  menos um. Para cada  $i$  tal que  $0 \leq i \leq x$ , o vetor  $s[i]$  representa o que um prisioneiro deve fazer se ler o número  $i$  na lousa ao entrar na sala:
  1. O valor de  $s[i][0]$  é 0 se o prisioneiro deve inspecionar o saco A ou 1 se o prisioneiro deve inspecionar o saco B.
  2. Seja  $j$  o número de moedas vistas no saco escolhido. O prisioneiro deve então realizar a seguinte ação:
    - Se o valor de  $s[i][j]$  for  $-1$ , o prisioneiro deve identificar o saco A como aquele que tem menos moedas.
    - Se o valor de  $s[i][j]$  for  $-2$ , o prisioneiro deve identificar o saco B como aquele que tem menos moedas.
    - Se o valor de  $s[i][j]$  for um número não-negativo, o prisioneiro deve escrever esse número na lousa. Note que  $s[i][j]$  deve ser, no máximo,  $x$ .
- Este procedimento é chamado exatamente uma vez.

## Exemplo

Considere a seguinte chamada:

```
devise_strategy(3)
```

Seja  $v$  o número que o prisioneiro lê na lousa ao entrar na sala. Uma das estratégias corretas é a seguinte:

- Se  $v = 0$  (incluindo o número inicial), inspecione o saco A.
  - Se ele contiver 1 moeda, identifique o saco A como aquele com menos moedas.
  - Se ele contiver 3 moedas, identifique o saco B como aquele que tem menos moedas.
  - Se ele contiver 2 moedas, escreva 1 na lousa (sobrescrevendo 0).
- Se  $v = 1$ , inspecione o saco B.
  - Se ele contiver 1 moeda, identifique o saco B como aquele com menos moedas.
  - Se ele contiver 3 moedas, identifique o saco A como aquele que tem menos moedas.
  - Se ele contiver 2 moedas, escreva 0 na lousa (sobrescrevendo 1). Note que este caso nunca pode acontecer, pois podemos concluir que ambos os sacos contêm 2 moedas, o que não é permitido.

Para reportar esta estratégia, o procedimento deve retornar  $[[0, -1, 1, -2], [1, -2, 0, -1]]$ . O tamanho do vetor retornado é 2, portanto, para este valor de retorno, o valor de  $x$  é  $2 - 1 = 1$ .

## Restrições

- $2 \leq N \leq 5000$

## Subtarefas

1. (5 pontos)  $N \leq 500$ , o valor de  $x$  não deve ser superior a 500.
2. (5 pontos)  $N \leq 500$ , o valor de  $x$  não deve ser superior a 70.
3. (90 pontos) O valor de  $x$  não deve ser superior a 60.

Se em qualquer um dos casos de teste, o vetor retornado por `devise_strategy` não representar uma estratégia correta, a pontuação de sua solução para essa sub tarefa será 0.

Na sub tarefa 3, você pode obter uma pontuação parcial. Seja  $m$  o valor máximo de  $x$  para os vetores retornados em todos os casos de teste nesta sub tarefa. Sua pontuação para esta sub tarefa é calculada de acordo com a tabela a seguir:

Condição	Pontos
$40 \leq m \leq 60$	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

## Corretor Exemplo

O corretor exemplo lê a entrada no seguinte formato:

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

Cada linha, exceto a primeira e a última, representa um cenário. Nos referimos ao cenário descrito na linha  $2 + k$  como cenário  $k$ . No cenário  $k$ , o saco A contém  $A[k]$  moedas e o saco B contém  $B[k]$  moedas.

O corretor exemplo chama primeiro `devise_strategy(N)`. O valor de  $x$  é o tamanho do vetor retornado pela chamada menos um. Então, se o corretor exemplo detectar que o vetor retornado por `devise_strategy` não está de acordo com as restrições descritas em Detalhes de Implementação, ele imprime uma das seguintes mensagens de erro e termina:

- `s is an empty array`:  $s$  é um vetor vazio (que não representa uma estratégia válida).
- `s[i] contains incorrect length`: Existe um índice  $i$  ( $0 \leq i \leq x$ ) tal que o tamanho de  $s[i]$  não é  $N + 1$ .
- `First element of s[i] is non-binary`: Existe um índice  $i$  ( $0 \leq i \leq x$ ) tal que  $s[i][0]$  não é nem 0 nem 1.
- `s[i][j] contains incorrect value`: Existem índices  $i, j$  ( $0 \leq i \leq x, 1 \leq j \leq N$ ) tais que  $s[i][j]$  não está entre  $-2$  e  $x$ .

Caso contrário, o corretor exemplo produz duas saídas.

Primeiro, o corretor exemplo imprime a saída de sua estratégia no seguinte formato:

- linha  $1 + k$  ( $0 \leq k$ ): a saída de sua estratégia para o cenário  $k$ . Se aplicar a estratégia leva a um prisioneiro identificar o saco A como aquele com menos moedas, então a saída é o

caractere A. Se aplicar a estratégia leva a um prisioneiro identificar o saco B como aquele com menos moedas, então a saída é o caractere B. Se aplicar a estratégia não leva nenhum prisioneiro a identificar um saco com menos moedas, então a saída é o caractere X.

Segundo, o corretor exemplo escreve um arquivo `log.txt` no diretório atual no formato a seguir:

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

A sequência na linha  $1 + k$  corresponde ao cenário  $k$  e descreve os números escritos na lousa. Especificamente,  $w[k][l]$  é o número escrito pelo  $(l + 1)$ -ésimo prisioneiro a entrar na sala.