Pyramids

Everyone knows that Pharaoh Khufu was a great ruler, but many are unaware that he was also a fashion enthusiast. Back in the day, he had n pyramids numbered from 0 to $n-1$, with pyramid i $(0 \leq i \leq n-1)$ having $a[i]$ stones. He also had the latest IKEA catalogue of the most fashionable pyramids of the year. The catalogue consists of n pyramids numbered from 0 to $n-1$, with pyramid i $(0 \leq i \leq n-1)$ having $b[i]$ stones.

Every day, Khufu would browse the catalogue and choose his favorite range of pyramids $[x, y]$ and then choose another range $[l, r]$ such that $r-l=y-x$. After that, he questions whether it's possible to transform his pyramids in range $[l, r]$ to appear like the catalogue's pyramids in range $[x, y]$. To transform the pyramids, he can move a stone from pyramid i to pyramid j $(l \leq i, j \leq r)$ any number of times. Notice that each time he can choose two different pyramids as long as they're inside the range $[l, r]$.

Your task is to determine whether it's possible to transform Khufu's pyramids in range $[l, r]$ to appear like the catalogue's pyramids in range $[x, y]$ for q days. Note that the number of stones in each pyramid never changes, Khufu only wonders if he can transform the pyramids but never changes them.

Implementation details

You should implement the following procedures:

```
void init(int[] a, int[] b)
```

- a, b: two arrays of length n, describing the number of stones in Khufu's and IKEA's pyramids respectively.
- This procedure is called exactly once, before any calls to `can_transform`.

```
bool can_transform(int l, int r, int x, int y)
```

- l, r: starting and ending indices of Khufu's pyramids range.
- x, y: starting and ending indices of IKEA's pyramids range.
- This procedure should return `true` if it's possible to transform the first subarray into the second and `false` otherwise.
- This procedure is called exactly q times, once for each day.

Example

Consider the following call:

```
init([1, 2, 3, 4, 5], [2, 2, 2, 4, 5])
```

Let's say the grader calls `can_transform(0, 2, 0, 2)`. This call should return whether we can transform Khufu's pyramids $[1, 2, 3]$ to $[2, 2, 2]$ or not. It is possible to transform it in by moving 1 stone from pyramid 2 to pyramid 0. Therefore, this call should return `true`.

Let's say the grader calls `can_transform(3, 4, 3, 4)`. This call should return whether we can transform Khufu's pyramids $[4, 5]$ to $[4, 5]$ or not. The pyramids already look alike. Therefore, this call should return `true`.

Let's say the grader calls `can_transform(0, 2, 1, 3)`. This call should return whether we can transform Khufu's pyramids $[1, 2, 3]$ to $[2, 2, 4]$ or not. It can be shown that there's no sequence of stone movements that achieves this transformation. Therefore, this call should return `false`.

Constraints

- $1 \le n, q \le 10^5$
- $1 \le a[i], b[i] \le 10^9$
- $0 \le l \le r \le n - 1$
- $0 \le x \le y \le n - 1$
- $r - l = y - x$

Subtasks

1. (10 points) $1 \le n$, $a[i]$, $b[i] \le 5$ and $1 \le q \le 10$.
2. (40 points) $1 \le n$, $q \le 1000$.
3. (20 points) $1 \le a[i], b[i] \le 2$.
4. (30 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

- line 1: n q
- line 2: $a[0]$ $a[1]$ $\ldots$ $a[n-1]$
- line 3: $b[0]$ $b[1]$ $\ldots$ $b[n-1]$
- line $4 + i$ (for all $0 \le i \le q - 1$): l r x y for the i-th call to `can_transform`.

The sample grader prints your answers in the following format:

- line $1 + i$ (for all $0 \leq i \leq q - 1$): the return value of the i-th call to `can_transform`.