



Sky Walking

Kenan hat einen Plan von Gebäuden und Verbindungsbrücken entlang einer Strassenseite der Hauptstrasse von Baku gezeichnet. Es gibt n Gebäude, die von 0 bis $n - 1$ nummeriert sind, und m Verbindungsbrücken, die von 0 bis $m - 1$ nummeriert sind. Der Plan ist zweidimensional, wobei die Gebäude als vertikale und die Verbindungsbrücken als horizontale Strecken gezeichnet sind. Es handelt sich also um eine Ansicht von vorne.

Das Fundament des Gebäudes i ($0 \leq i \leq n - 1$) befindet sich an der Stelle $(x[i], 0)$ und das Gebäude hat eine Höhe $h[i]$. Es ist daher durch eine Strecke dargestellt, die die Punkte $(x[i], 0)$ und $(x[i], h[i])$ verbindet.

Die Verbindungsbrücke j ($0 \leq j \leq m - 1$) hat Endpunkte an den Gebäuden, die mit $l[j]$ and $r[j]$ bezeichnet sind, und hat eine positive y -Koordinate $y[j]$. Sie ist daher durch die Strecke dargestellt, die die Punkte $(x[l[j]], y[j])$ and $(x[r[j]], y[j])$ verbindet.

Eine Verbindungsbrücke und ein Gebäude **schneiden** sich, wenn sie einen gemeinsamen Punkt besitzen. Eine Verbindungsbrücke schneidet somit zwei Gebäude in ihren Endpunkten und kann auch andere Gebäude dazwischen schneiden.

Kenan möchte die Länge des kürzesten Weges vom Fundament des Gebäudes s zum Fundament des Gebäudes g finden oder herausfinden, dass kein solcher Weg existiert. Dabei kann er nur über Gebäude und Verbindungsbrücken gehen. Beachte, dass es nicht erlaubt ist, am Boden zu gehen, das heisst entlang der horizontalen Linie mit der y -Koordinate 0.

Man darf bei jeder Überschneidung von einer Verbindungsbrücke in ein Gebäude und umgekehrt gehen. Wenn die Endpunkte von zwei Verbindungsbrücken zusammenfallen, kann man direkt von einer Verbindungsbrücke zur anderen gehen.

Hilf Kenan bei der Beantwortung dieser Frage.

Implementierung

Implementiere die folgende Funktion. Sie wird vom Grader einmal pro Testfall aufgerufen.

```
int64 min_distance(int[] x, int[] h, int[] l, int[] r, int[] y,  
                  int s, int g)
```

- x und h : Integer-Arrays der Länge n
- l , r und y : Integer-Arrays der Länge m
- s und g : zwei Integer
- Diese Funktion soll die Länge des kürzesten Pfades zwischen den Böden der Gebäude s und g zurückgeben, falls ein solcher Pfad existiert. Sonst soll sie -1 zurückgeben.

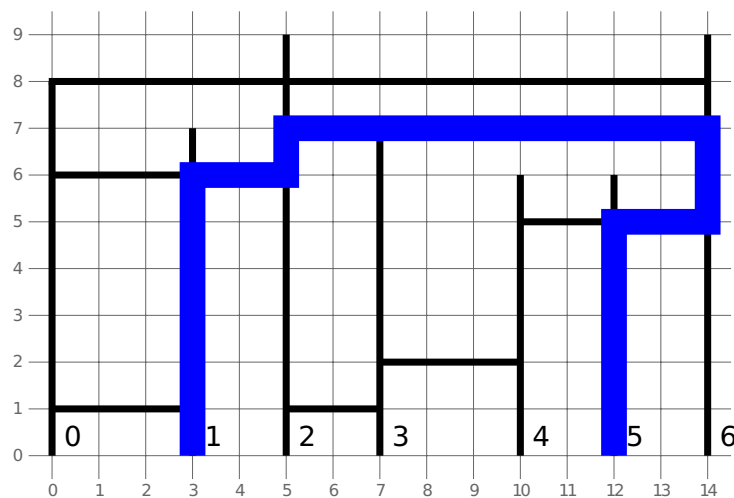
Beispiele

Beispiel 1

```
min_distance([0, 3, 5, 7, 10, 12, 14],
            [8, 7, 9, 7, 6, 6, 9],
            [0, 0, 0, 2, 2, 3, 4],
            [1, 2, 6, 3, 6, 4, 6],
            [1, 6, 8, 1, 7, 2, 5],
            1, 5)
```

Die richtige Antwort ist 27.

Die folgende Abbildung gehört zum *Beispiel 1*:



Beispiel 2

```
min_distance([0, 4, 5, 6, 9],
            [6, 6, 6, 6, 6],
            [3, 1, 0],
            [4, 3, 2],
            [1, 3, 6],
            0, 4)
```

Die richtige Antwort ist 21.

Beschränkungen

- $1 \leq n, m \leq 100\,000$
- $0 \leq x[0] < x[1] < \dots < x[n-1] \leq 10^9$
- $1 \leq h[i] \leq 10^9$ (für alle $0 \leq i \leq n-1$)
- $0 \leq l[j] < r[j] \leq n-1$ (für alle $0 \leq j \leq m-1$)
- $1 \leq y[j] \leq \min(h[l[j]], h[r[j]])$ (für alle $0 \leq j \leq m-1$)
- $0 \leq s, g \leq n-1$
- $s \neq g$
- Keine zwei Verbindungsbrücken haben einen gemeinsamen Punkt, ausser möglicherweise an ihren Endpunkten.

Subtasks

1. (10 Punkte) $n, m \leq 50$
2. (14 Punkte) Jede Verbindungsbrücke schneidet höchstens 10 Gebäude.
3. (15 Punkte) $s = 0, g = n-1$ und alle Gebäude haben die gleiche Höhe.
4. (18 Punkte) $s = 0, g = n-1$
5. (43 Punkte) Keine zusätzlichen Beschränkungen.

Beispiel-Grader

Der Beispiel-Grader liest die Eingaben in folgendem Format:

- Zeile 1: $n \ m$
- Zeile $2 + i$ ($0 \leq i \leq n-1$): $x[i] \ h[i]$
- Zeile $n + 2 + j$ ($0 \leq j \leq m-1$): $l[j] \ r[j] \ y[j]$
- Zeile $n + m + 2$: $s \ g$

Der Beispiel-Grader gibt eine einzigen Zeile mit dem Rückgabewert von `min_distance` aus.