



Sky Walking

Kenan heeft een kaart gemaakt van de gebouwen en skywalks aan één kant van de hoofdstraat van Baku. Er zijn n gebouwen, genummerd van 0 tot en met $n - 1$ en m skywalks, genummerd van 0 tot en met $m - 1$. De kaart is getekent op een tweedimensionaal vlak waar de gebouwen en skywalks horizontaal respectievelijk verticaal zijn getekend.

De onderkant van gebouw i ($0 \leq i \leq n - 1$) ligt op positie $(x[i], 0)$, en het gebouw heeft een hoogte $h[i]$. Dus, het is een segment dat de punten $(x[i], 0)$ en $(x[i], h[i])$ met elkaar verbindt.

Skywalk j ($0 \leq j \leq m - 1$) heeft eindpunten op de gebouwen $l[j]$ en $r[j]$ en heeft een positief y -coördinaat, $y[j]$. Dus, het is een segment dat de punten $(x[l[j]], y[j])$ en $(x[r[j]], y[j])$ met elkaar verbindt.

Een skywalk en een gebouw **snijden** als ze een punt gemeen hebben. Dus, een skywalk snijdt de beide gebouwen op de eindpunten, en kan ook gebouwen die ertussen staan snijden.

Kenan probeert de lengte van het kortste pad van de onderkant van gebouw s naar de onderkant van gebouw g te achterhalen, of te bepalen dat dit pad niet bestaat. Je kunt alleen wandelen door gebouwen en skywalks. Let op dat je niet over de grond mag wandelen, je mag dus niet over de horizontale as met y -coördinaat 0 wandelen.

Je kunt vanaf een skywalk een gebouw ingaan, en andersom, bij elk snijpunt. Als de snijpunten van twee skywalks op hetzelfde punt liggen dan kun je ook van de ene skywalk naar de andere lopen.

Help Kenan zijn vraag te beantwoorden.

Implementatiedetails

Implementeer de volgende procedure. Deze wordt door de grader exact één keer aangeroepen voor elke test case.

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

- x en h : arrays van lengte n met gehele getallen.

- l , r , en y : arrays van lengte m met gehele getallen.
- s en g : twee gehele getallen.
- Deze procedure moet de lengte van het kortste pad tussen de onderkant van gebouw s en de onderkant van gebouw g teruggeven, als dit pad bestaat. Als het pad niet bestaat moet de functie -1 teruggeven.

Voorbeelden

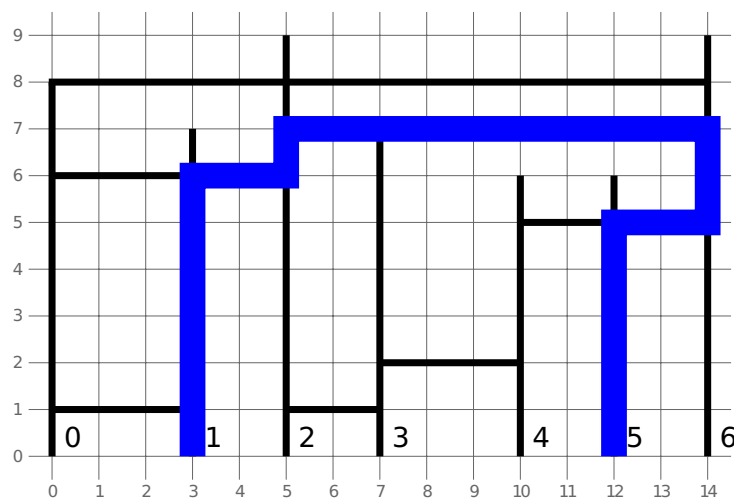
Voorbeeld 1

Kijk naar de volgende aanroep:

```
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)
```

Het juiste antwoord is 27.

De volgende figuur hoort bij *voorbeeld 1*.



Voorbeeld 2

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

Het juiste antwoord is 21.

Randvoorwaarden

- $1 \leq n, m \leq 100\,000$
- $0 \leq x[0] < x[1] < \dots < x[n-1] \leq 10^9$
- $1 \leq h[i] \leq 10^9$ (voor alle $0 \leq i \leq n-1$)
- $0 \leq l[j] < r[j] \leq n-1$ (voor alle $0 \leq j \leq m-1$)
- $1 \leq y[j] \leq \min(h[l[j]], h[r[j]])$ (voor alle $0 \leq j \leq m-1$)
- $0 \leq s, g \leq n-1$
- $s \neq g$
- Er zijn geen twee skywalks die een punt gemeen hebben, behalve wellicht op de eindpunten.

Subtasks

1. (10 punten) $n, m \leq 50$
2. (14 punten) Elke skywalk snijdt maximaal 10 gebouwen.
3. (15 punten) $s = 0, g = n-1$ en alle gebouwen zijn even hoog.
4. (18 punten) $s = 0, g = n-1$.
5. (43 punten) Geen aanvullende voorwaarden.

Voorbeeld grader

De voorbeeld grader leest de invoer in het volgende formaat:

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

De voorbeeld grader print een enkele regel met daarop het resultaat van de aanroep van `min_distance`.