Абадагы көпүрөлөр

Кенан Баку шаарынын башкы көчөсүндө жайгашкан имараттардын жана Абадагы көпүрөлөрдүн долбоорун чызды. Ал жерде n Имарат 0дөн $(n-1)$ге чейин жана m Абадагы көпүрө 0дөн $(m-1)$ге чейин номерленген. Эки өлчөмдүү мейкиндикте Имараттар вертикалдуу жана абадагы көпүрөлөр горизонталдуу кесиндилер.

i инчи имараттын түбү $(0 \le i \le n-1)$ $(x[i], 0)$ точкасында жайгашкан, жана бийиктиги $h[i]$. Ошондуктан имарат $(x[i], 0)$ жана $(x[i], h[i])$ точкаларды бириктирген кесинди.

j инчи $(0 \le j \le m-1)$ абадагы көпүрөнүн эки учу $l[j]$инчи жана $r[j]$инчи имараттарга байланган жана анын y-координатасы $y[j]$ (оң сан). Ошондуктан,абадагы көпүрөлөр $(x[l[j]], y[j])$ жана $(x[r[j]], y[j])$ точкаларды бириктирген кесинди.

Эгерде абадагы көпүрө менен имарат жалпы точкага ээ болсо анда алар **уланган** деп аталат.

Ошондуктан , абадагы көпүрө эки имаратка уланган жана экөөнун арасындагы башка имаратка уланган болушу мүмкүн.

Кенан s инчи имараттын түбүнөн g имараттын түбүнө эң кыска жолдун узундугун тапкысы келет, бара албаса жол жок экенин айтуусу керек.

Жерде басууга болбойт.

(Абадагы көпүрөдөн имаратка же башка абадагы көпүрөгө) же (имараттан абадагы көпүрөгө) каалаган **уланган** чекиттен өтсө болот.

Кенанга жардам бергиле

Implementation details

You should implement the following procedure. It will be called by the grader once for each test case.

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

- x and h: integer arrays of length n

- l, r, and y: integer arrays of length m
- s and g: two integers
- This procedure should return the length of the shortest path between the bottom of building s and the bottom of building g, if such path exists. Otherwise, it should return -1.

Examples

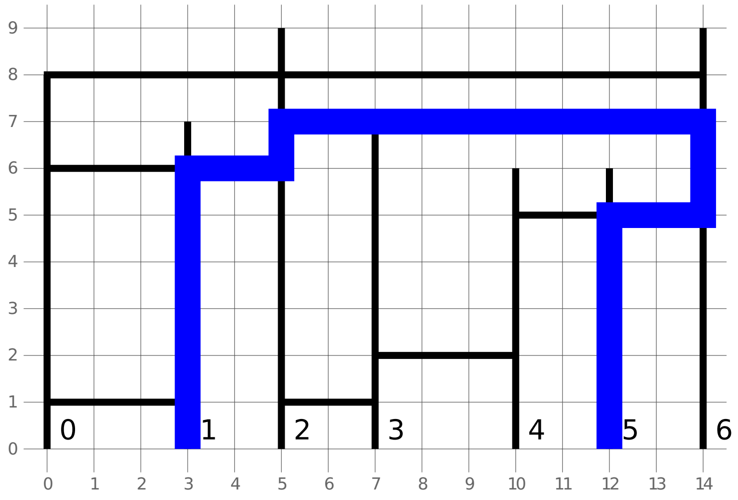
Example 1

Consider the following call:

```
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)
```

The correct answer is 27.

The figure below corresponds to *Example 1*:



Example 2

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

The correct answer is 21.

Constraints

- $1 \le n, m \le 100\,000$
- $0 \le x[0] < x[1] < \ldots < x[n-1] \le 10^9$
- $1 \le h[i] \le 10^9$ (for all $0 \le i \le n-1$)
- $0 \le l[i] < r[i] \le n-1$ (for all $0 \le i \le m-1$)
- $1 \le y[i] \le \min(h[l[i]], h[r[i]])$ (for all $0 \le i \le m-1$)
- $0 \le s, g \le n-1$
- $s \ne g$
- No two skywalks have a common point, except maybe on their endpoints.

Subtasks

1. (10 points) $n, m \le 50$
2. (14 points) Each skywalk intersects at most 10 buildings.
3. (15 points) $s = 0$, $g = n-1$, and all buildings have the same height.
4. (18 points) $s = 0$, $g = n-1$
5. (43 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

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

The sample grader prints a single line containing the return value of `min_distance`.