

IOI '19 P6 - Sky Walking

Time limit: 1.8s **Memory limit:** 1G

Kenan drew a plan of the buildings and skywalks along one side of the main avenue of Baku. There are n buildings numbered from 0 to $n - 1$ and m skywalks numbered from 0 to $m - 1$. The plan is drawn on a two-dimensional plane, where the buildings and skywalks are vertical and horizontal segments respectively.

The bottom of building i ($0 \leq i \leq n - 1$) is located at point $(x[i], 0)$ and the building has height $h[i]$. Hence, it is a segment connecting the points $(x[i], 0)$ and $(x[i], h[i])$.

Skywalk j ($0 \leq j \leq m - 1$) has endpoints at buildings numbered $l[j]$ and $r[j]$ and has a positive y -coordinate $y[j]$. Hence, it is a segment connecting the points $(x[l[j]], y[j])$ and $(x[r[j]], y[j])$.

A skywalk and a building **intersect** if they share a common point. Hence, a skywalk intersects two buildings at its two endpoints, and may also intersect other buildings in between.

Kenan would like to find the length of the shortest path from the bottom of building s to the bottom of building g , assuming that one can only walk along the buildings and skywalks, or determine that no such path exists. Note that it is not allowed to walk on the ground, i.e. along the horizontal line with y -coordinate 0.

One can walk from a skywalk into a building or vice versa at any intersection. If the endpoints of two skywalks are at the same point, one can walk from one skywalk to the other.

Your task is to help Kenan answer his question.

Implementation details

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

```
long long min_distance(std::vector<int> x, std::vector<int> h, std::vector<int> l,
std::vector<int> r, std::vector<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 and the bottom of building g , if such path exists. Otherwise, it should return -1 .

Examples

Example 1

Consider the following call:

- No two skywalks have a common point, except maybe on their endpoints.

Subtasks

1. (10 points) $n, m \leq 50$
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 $2 + i$ ($0 \leq i \leq n - 1$): $x[i] h[i]$
- line $n + 2 + j$ ($0 \leq j \leq m - 1$): $l[j] r[j] y[j]$
- line $n + m$: $s g$

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