



Sky Walking

O Kenan desenhou um plano dos edifícios e das pontes aéreas entre edifícios ao longo de um lado da avenida principal de Baku. Há n edifícios numerados de 0 a $n - 1$ e m pontes aéreas numeradas de 0 a $m - 1$. O plano está desenhado num plano bi-dimensional, onde os edifícios e pontes aéreas entre eles são segmentos de reta horizontais e verticais respetivamente.

O chão do edifício i ($0 \leq i \leq n - 1$) está localizado no ponto $(x[i], 0)$ e o edifício tem altura $h[i]$. Assim, é um segmento que liga os pontos $(x[i], 0)$ e $(x[i], h[i])$.

A ponte aérea j ($0 \leq j \leq m - 1$) tem as extremidades nos edifícios numerados $l[j]$ e $r[j]$ e tem uma coordenada $y, y[j]$, positiva. Assim, é um segmento que liga os pontos $(x[l[j]], y[j])$ e $(x[r[j]], y[j])$.

Uma ponte aérea e um edifício **interseitam-se** se partilham um ponto em comum. Assim, uma ponte aérea intersesta dois edifícios nas suas extremidades, e pode também intersestar outros edifícios entre estas.

O Kenan gostaria de encontrar o comprimento do caminho mais curto do chão do edifício s para o chão do edifício g , assumindo que só pode andar por cima dos edifícios e das pontes aéreas. Nota que não é permitido andar pelo chão, isto é, ao longo da linha horizontal com coordenada y igual a 0.

Pode-se andar de uma ponte aérea para um edifício, e vice versa, em qualquer interseção. Se as extremidades de duas pontes aéreas forem o mesmo ponto, então pode andar de uma para a outra.

A tua tarefa é ajudar o Kenan a responder a esta pergunta.

Detalhes de Implementação

Deves implementar a seguinte função. Vai ser chamada pelo avaliador uma vez para cada caso de teste.

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

- x e h : arrays de inteiros de comprimento n
- $l, r,$ e y : arrays de inteiros de comprimento m

- s e g : dois inteiros
- Esta função deve devolver o comprimento do caminho mais curto entre o chão do edifício s e o chão do edifício g , se tal caminho existir. Caso contrário, deve devolver -1 .

Exemplos

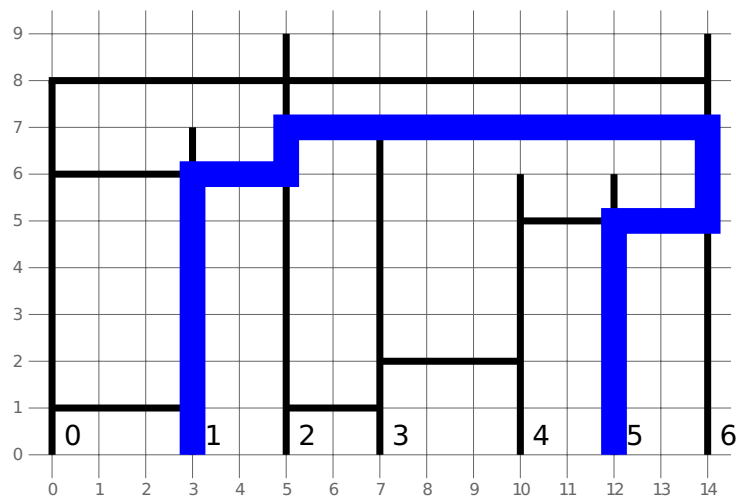
Exemplo 1

Considera a seguinte chamada:

```
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)
```

A resposta correta é 27.

A figura abaixo corresponde ao *Exemplo 1*:



Exemplo 2

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

A resposta correta é 21.

Restrições

- $1 \leq n, m \leq 100\,000$
- $0 \leq x[0] < x[1] < \dots < x[n-1] \leq 10^9$
- $1 \leq h[i] \leq 10^9$ (para todos $0 \leq i \leq n-1$)
- $0 \leq l[i] < r[i] \leq n-1$ (para todos $0 \leq i \leq m-1$)
- $1 \leq y[i] \leq \min(h[l[i]], h[r[i]])$ (para todos $0 \leq i \leq m-1$)
- $0 \leq s, g \leq n-1$
- $s \neq g$
- Nenhum par de pontes aéreas têm um ponto em comum, exceto talvez nas suas extremidades.

Subtarefas

1. (10 pontos) $n, m \leq 50$
2. (14 pontos) Cada ponte aérea intersesta no máximo 10 edifícios.
3. (15 pontos) $s = 0, g = n-1$, e todos os edifícios têm a mesma altura.
4. (18 pontos) $s = 0, g = n-1$
5. (43 pontos) Nenhuma restrição adicional.

Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato:

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

O avaliador exemplo imprime uma única linha que contém o valor de retorno de `min_distance`.