Reservas de monorriel

Contribución de Guillermo García

Descripción del problema

Para viajar hacia el centro de una ciudad, parte todas las mañanas un tren monorriel desde la estación terminal situada en las afueras. Realiza varias paradas intermedias en distintas estaciones, antes de llegar a la estación terminal situada en el centro. Son en total **N** estaciones incluyendo ambas terminales, que se numeran en el orden del recorrido desde **1** para la terminal en las afueras, hasta **N** para la terminal en el centro.

La demanda de pasajes es muy alta, y el tren tiene una capacidad máxima de **C** pasajeros. Esto significa que nunca puede haber en el tren más que **C** personas. Un detalle importante es que en cada estación, **primero descienden** del tren todas las personas que desean bajar allí, y **luego ascienden** todas las personas que quieren subir al tren en esa misma estación.

Por esta razón, la ciudad desea implementar un sistema de reservas computarizado: el día previo a viajar, hasta las 18:00 horas, cada uno de los **M** pasajeros interesados realizará una reserva indicando en cuál estación se subirá al tren, y en cuál estación descenderá.

A las 19:00 horas se enviará una respuesta a cada pasajero indicándole si su reserva está confirmada o no. Tu tarea consiste en implementar el algoritmo que confirma las reservas, con el único objetivo de que viajen la mayor cantidad de personas posibles.

Descripción de la función

Debes implementar la función `monorriel(C, a, b, reservas)`. Sus parámetros son:

- `C`: un entero que indica la capacidad del tren.
- `a,b`: arreglos de **M** enteros entre **1** y **N**, que indican las solicitudes de reservas. El **i**-ésimo pasajero quiere subir al tren en la estación **a**[**i**], y bajar en la **b**[**i**].
- `reservas`: Arreglo en el que escribir **M** enteros, que serán ceros y unos, indicando las reservas. Para confirmar la reserva del **i**-ésimo pasajero, **reservas**[**i**] debe ser **1**, y en caso contrario debe ser **0**.

La función debe retornar un entero, que indique la máxima cantidad de reservas que es posible confirmar. En caso de haber más de una forma de obtener esta máxima cantidad, se puede escribir cualquiera de ellas en el arreglo `reservas`.

Evaluador

El evaluador local lee de la entrada estándar con el siguiente formato:

- Primera línea: dos enteros **C** y **M**
- Luego **M** líneas, cada una con dos enteros **a**[**i**] y **b**[**i**]

El evaluador local escribe a la salida estándar una primera línea con el valor retornado por la función, y una segunda línea con el contenido del arreglo `reservas`.

Restricciones

- $1 \leq a_i < b_i \leq 10^9$
- $1 \leq C, M \leq 200.000$

Ejemplo

Si se invoca al evaluador con la siguiente entrada:

```
3 5
1 10
2 15
4 20
3 17
11 12
```

Para un programa correcto, una posible salida sería:

```
4
1 0 1 1 1
```

Si en cambio fuera:

```
1 2
1 10
10 15
```

La salida sería:

```
2
1 1
```

Puntuación

Se obtiene 50 % del puntaje por el correcto valor de retorno de la función, y el 50 % restante por además dar una respuesta correcta en el arreglo `reservas`.

Subtareas

1. **N** = **3** (10 ptos.)
2. **C** = **1** (12 ptos.)
3. **M** $\leq$ **1000** (22 ptos.)
4. **N** = **1000** (26 ptos.)
5. Sin más restricción (30 ptos.)