Bicing bifurcado

PRO2 Otoño 2023

1 Introducción

Tenemos un sistema de Bicing con una estación especial, que llamaremos "primera estación". Cada estación tiene dos o cero estaciones siguientes.

Cada estación tiene un identificador string que solo tiene letras y digítos. Cada estación puede albergar un número diferente de bicis. Cada estación ha de saber qué bicis tiene. Se puede cambiar la capacidad de una estación.

Cada bicicleta tiene un identificador string que solo tiene letras y digítos. Una bici ha de saber qué viajes ha hecho. Una bici está siempre asignada a una estación, que puede ir cambiando.

Se comienza leyendo la estructura de las estaciones de bicing junto con la capacidad de cada una.

A partir de este punto se puede realizar una serie de acciones que involucran estaciones y bicis:

- Se puede dar de alta, en una estación concreta, una bicicleta que no existe.
- Se puede dar de baja una bici que existe.
- Se pueden listar los datos de una bici, es decir, en qué estación está y qué viajes ha hecho.
- Se puede mover una bici concreta desde la estación a la que está asignada a otra estación.
- Se pueden listar las bicis de una estación, por orden de identificador.
- Se puede modificar la capacidad de una estación.
- Se puede consultar el total de plazas libres.

Además hay otras acciones más complicadas porque involucran todas las estaciones.

1.1 Reestructurar ubicación

Ocasionalmente se reestructurará la ubicación de las bicis, moviéndolas en dirección a la primera estación. Se intentará llenar una estación con bicis de las dos estaciones siguientes. Cogeremos las bicis que hagan falta para llenarla (si es posible) de manera que las dos subestaciones se queden con un número lo más equilibrado posible de bicis. Moveremos las que tengan identificador más pequeño. La primera estación está en el nivel 1. Se comienza moviendo bicis de las dos estaciones del nivel 2 (si existen) a la estación del nivel 1. Dada una estación no se pueden mover bicis de las dos estaciones siguientes a ella si antes no se ha hecho con todas las estaciones antecesoras.

1.2 Asignar estación

También se puede asignar una estación a una bici dónde se considere que hace más falta añadir bicis nuevas. Para ello se escoge la estación que tenga mayor coeficiente de desocupación general, que se define como el número total de plazas sin usar de la estación y estaciones siguientes dividido entre el número de estaciones a partir de la misma, incluida. En caso de empate se escoge la estación con identificador menor. Una vez escogida dicha estación, la bici se da de alta en dicha estación.

2 Operaciones

Todos los identificadores, sean de estaciones o bicis, tienen el mismo formato: strings que solo tiene letras y digítos.

La capacidad de una estación será siempre mayor que cero.

Los detalles exactos del formato de la entrada y la salida se han de consultar en el juego de pruebas público del jutge.

Hay que comprobar los posibles errores de cada operación en el orden en que se listan puesto que en varias operaciones pueden producirse varios errores incluso simultáneamente.

1. **alta_bici** o **ab**. Se leerá un identificador de bici y un identificador de estación. Si la bici existe se produce un mensaje de error. Si la estación no existe se produce un mensaje de error. Si la bici no cabe en la estación se produce un mensaje de error. Si no se produce ningún error la bici pasa a estar en dicha estación. No se escribe nada.

2. **baja_bici** o **bb**. Se leerá un identificador de bici. Si la bici no existe se produce un mensaje de error. Si no se produce ningun error, la bici desaparece del sistema. No se escribe nada.

3. **estacion_bici** o **eb**. Se leerá un identificador de bici. Si la bici no existe se produce un mensaje de error. Si la bici existe, necesariamente ha de estar en una estación y se escribe el identificador de dicha estación.

4. **viajes_bici** o **vb**. Se leerá un identificador de bici. Si la bici no existe se produce un mensaje de error. Si no se produce un error se escribirán los viajes que ha hecho la bici.

5. **mover_bici** o **mb**. Se leerá un identificador de bici y un identificador de estación que corresponderá a la estación de destino. Si la bici no existe se produce un mensaje de error. Si la bici existe, está en una estación. Si la estación de destino no existe se produce un mensaje de error. Si la estación de destino es en la que está la bici se produce un mensaje de error. Si la bici no cabe en la estación de destino se produce un mensaje de error. Si no se produce ningún error se añade a la bici un viaje desde la estación de origen a la de destino y la bici pasa a estar en la estación de destino.

6. **bicis_estacion** o **be**. Se leerá un identificador de estación. Si la estación no existe se produce un mensaje de error. Si la estación existe, se dirá qué bicis tiene, por orden de identificador.

7. **modificar_capacidad** o **mc**. Se leerá un identificador de estación y un número estrictamente positivo. Si la estación no existe se produce un mensaje de error. Si el número es menor que la cantidad de bicis que hay en la estación se produce un mensaje de error. Si no se producen errores se cambia el número de bicis que puede acoger la estación.

8. **plazas_libres** o **pl**. Nos dice cuantas plazas libres quedan entre todas las estaciones. No tiene parámetros ni produce errores.

9. **subir_bicis** o **sb**. Acerca bicis hacia la primera estación según se indica en 1.1. No tiene parámetros ni produce errores, ni escribe nada.

10. **asignar_estacion** o **ae**. Se leerá un identificador de bici. Si la bici existe se produce un mensaje de error. Si la bici no cabe en ninguna estación se produce un mensaje de error. Si no se producen errores, la bici queda asignada a la estación escogida siguiendo los criterios explicados en 1.2 y se escribe el identificador de la estación.

11. **fin** Acaba la ejecución.