

Jugando con cupones

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

Agustín y Carlos juegan habitualmente un juego de dos jugadores por turnos. Cada vez que se juntan a jugar, ponen sobre la mesa todos los *cupones* que tienen. Cada cupón tiene escrito un cierto número entero positivo que indica su *valor*.

Siempre empieza jugando Agustín y en su primer turno, puede elegir cualquier cupón de la mesa, retirarlo de allí y guardarlo en su mano. Luego es el turno de Carlos, que puede elegir cualquiera de los restantes cupones sobre la mesa y llevárselo a su mano. Se sigue de esta misma manera, con ambos jugadores alternando turnos, y cada jugador en su turno elige algún cupón de los que quedan en la mesa y se lo lleva a su mano.

El juego termina cuando ya no quedan cupones sobre la mesa. En ese momento, el puntaje de cada jugador es la suma de los valores de los cupones que tiene en su mano.

Como jugar siempre con los mismos cupones es aburrido, la colección de cupones que utilizan Agustín y Carlos va cambiando. Inicialmente esta colección de cupones es vacía, y con el paso del tiempo se van agregando o sacando cupones.

Debes escribir dos funciones que procesen cada evento de agregar o sacar un cupón, y calculen luego de realizado ese cambio, cuál es el máximo puntaje que puede obtener Agustín si ambos juegan de forma óptima con los cupones de la colección en ese momento. Cada jugador intenta obtener para sí mismo el máximo puntaje posible.

Descripción de la función

Se deben escribir dos funciones `agregar(x)` y `sacar(x)`. En ambos casos, el parámetro x indica el valor del cupón que se agrega o se saca de la colección respectivamente. En el caso de `sacar(x)`, se garantiza que cuando se llama a la función efectivamente existirá al menos un cupón de valor x para sacar.

En ambos casos la función debe retornar un único entero: el puntaje de Agustín al terminar el juego con todos los cupones de la colección que resulta luego de agregar o sacar el cupón indicado, asumiendo que Agustín y Carlos juegan de manera óptima.

Evaluador local

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

- Una línea con un entero Q
- Q líneas, con una letra s o a y un número x

El evaluador local realiza Q llamadas a función en el orden de la entrada: por cada línea de la forma $a\ x$ realiza una llamada `agregar(x)`, y por cada línea de la forma $s\ x$ realiza una llamada `sacar(x)`.

El evaluador local escribe en la salida estándar Q líneas, que indican los valores retornados por cada llamada a función.

Restricciones

$$1 \leq Q \leq 200\,000$$

$$1 \leq x \leq 10^{13}$$

Ejemplo

Si el evaluador local recibe:

```
7
a 10
a 15
s 15
s 10
a 5
a 5
a 6
```

La salida correcta es:

```
10
15
10
0
5
5
11
```

Subtareas

1. $Q = 2$ (1 punto)
2. $Q = 3$ (2 puntos)
3. $Q \leq 1000$ (4 puntos)
4. $x \leq 2$ (8 puntos)
5. $x \leq 10$ (9 puntos)
6. $x \leq 200\,000$ (19 puntos)
7. Solamente llamadas a la función `agregar`, y se garantiza que el orden de la secuencia de Q llamadas `agregar(x)` es una permutación de esas llamadas elegida al azar uniformemente (por ejemplo con la función `random_shuffle`) (20 puntos)
8. Solamente llamadas a la función `agregar` (22 puntos)
9. Sin más restricción (15 puntos)