

## Emparejando jugadores de Age of Empires

Contribución de Facundo Gutiérrez

### Descripción del problema

A Facu le gusta jugar al Age of Empires, uno de los mejores videojuegos de estrategia en tiempo real de todos los tiempos. En cada partida se enfrentan dos jugadores entre sí.

En este videojuego online existe un sistema automático de emparejamiento de jugadores. Cada jugador tiene un cierto valor de habilidad, comunmente denominado su *elo*, en honor al creador del sistema, el físico húngaro Arpad Elo. En el Age of Empires, el elo de un jugador es un entero positivo.

El sistema de emparejamientos conoce los elos de todos los jugadores, y prioriza emparejar a cada jugador con un rival de elo similar. Facu considera que el sistema puede mejorarse, y te encarga la tarea de implementar el sistema de emparejamientos mejorado. El funcionamiento preciso del sistema de emparejamientos que debes implementar se describe a continuación.

El sistema funciona manteniendo siempre un conjunto de jugadores en espera, que está inicialmente vacío. Para evitar hacer esperar demasiado a los jugadores y causar que abandonen el juego, el sistema permite emparejar en cualquier momento a dos jugadores  $A$  y  $B$  que estén en espera aunque tengan distinto elo, siempre y cuando **alguno** de los dos haya estado en espera al menos **tantos segundos como su diferencia de elo**. A este criterio ingenioso se lo denomina la *regla de la diferencia*.

Por ejemplo, si el jugador  $A$  tiene un elo  $e_A = 1500$  y el jugador  $B$  tiene un elo  $e_B = 1450$ , el sistema permitirá emparejarlos si **al menos uno** de ellos ha estado esperando ya durante 50 segundos.

Cuando un jugador  $A$  desea buscar un oponente online para jugar, ingresa al sistema y hace clic en un botón para buscar rival.

- Si ya existe en espera algún jugador  $B$  de **exactamente** el mismo elo que  $A$ , el sistema empareja **inmediatamente** a los jugadores  $A$  y  $B$ , que iniciarán su partida y por lo tanto ya no estarán a la espera ni serán considerados por el sistema en adelante.
- Si no existe un jugador del mismo elo, pero existe otro jugador  $B$  que puede emparejarse con  $A$  inmediatamente de acuerdo a la regla de la diferencia, el sistema empareja inmediatamente a los jugadores  $A$  y  $B$ .
- En caso contrario, el jugador  $A$  se incorpora al conjunto de jugadores en espera.

Emparejar jugadores que estaban esperando siempre tiene prioridad por sobre incorporar jugadores nuevos: si un jugador hace clic en el botón para buscar rival en un cierto instante de tiempo, y en ese mismo instante es posible de acuerdo a la regla de la diferencia emparejar a dos jugadores que ya estaban esperando, el sistema realiza emparejamientos entre los jugadores en espera mientras sea posible, antes de procesar la incorporación jugadores nuevos.

Si en un instante de tiempo se habilitan simultáneamente dos o más posibles pares de jugadores para emparejar de acuerdo a la regla de la diferencia, el sistema siempre elegirá realizar un emparejamiento que involucre a un jugador con un elo lo más alto posible. Se puede demostrar que este criterio ya es suficiente para evitar ambigüedad a la hora

de seleccionar un emparejamiento entre varios posibles, cuando se implementa el resto del sistema de emparejamientos correctamente.

Si varios jugadores diferentes hacen clic en el botón de buscar rival exactamente al mismo tiempo, el sistema los procesa de uno en uno siguiendo estas reglas, en el orden en que estos eventos sean recibidos por el programa.

Tu tarea consiste en indicar los emparejamientos que va generando el sistema, a medida que se te van reportando los eventos que van ocurriendo desde el estado inicial con el conjunto de jugadores en espera vacío. Estos eventos serán de dos tipos:

1. Transcurren exactamente  $t$  segundos.
2. Un nuevo jugador con un determinado elo  $e$  hace clic en el botón de buscar rival.

Se asume que el sistema de procesamiento es instantáneo, es decir, el tiempo solamente transcurre para los cálculos de la regla de la diferencia, cuando se te informa un evento de tipo 1.

## Descripción de la función

Deberás escribir tres funciones:

`inicializar(N)`, será llamada al comenzar el programa, y puedes utilizarla para inicializar tus estructuras de datos globales si así lo deseas. Recibe la cantidad  $N$  de eventos a procesar, es decir la cantidad de llamadas totales a las funciones `tiempo` y `jugador`.

`tiempo(t)`, que será llamada para indicar que transcurren  $t$  segundos.

`jugador(e)`, que será llamada cada vez que un nuevo jugador busque rival. Recibe un entero  $e$  que indica el elo del jugador que hizo clic en el botón de buscar rival.

Las funciones `tiempo` y `jugador` deben retornar un arreglo de enteros que indique todos los emparejamientos que han ocurrido luego de procesar el evento correspondiente, en el orden en que ocurren. Si ocurren  $K$  emparejamientos, se retorna un arreglo  $a$  con  $2K$  enteros: el  $i$ -ésimo emparejamiento (para  $0 \leq i < K$ ) enfrenta un jugador de elo  $a[2*i]$  con un jugador de elo  $a[2*i+1]$ . Deberá ser  $a[2*i] \leq a[2*i+1]$ .

Cuando en el mismo instante de tiempo más de un jugador hace clic en el botón de buscar rival, las llamadas a `jugador` serán realizadas en el orden en que estos eventos son recibidos por el programa, es decir, en el orden en que deben ser procesados.

## Evaluador local

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

- Una línea con un entero  $N$
- $N$  líneas más, la  $i$ -ésima de las cuales contiene dos enteros  $b_i, x_i$

El evaluador local primero realiza la llamada `inicializar(N)`, y luego en orden las  $N$  llamadas a las otras funciones.

Si  $b_i = 1$ , la  $i$ -ésima llamada es `tiempo(x_i)`

Si  $b_i = 2$ , la  $i$ -ésima llamada es jugador ( $x_i$ )

Luego de cada llamada, escribe en la salida estándar una línea que representa el arreglo retornado en la llamada: si se retorna un arreglo de tamaño  $K$ , el evaluador escribe el entero  $K$  seguido de los  $K$  enteros del arreglo.

### Restricciones

$$1 \leq N \leq 100.000$$

$$1 \leq x_i \leq 10^9 = 1.000.000.000$$

$$b_i \in \{1, 2\}$$

### Ejemplo

Si el evaluador local recibe:

```
10
2 1500
1 1
2 1600
1 98
2 500
1 5
2 506
2 507
2 508
1 100
```

La salida correcta es:

```
0
0
0
0
0
2 1500 1600
0
0
0
4 507 508 500 506
```

### Subtareas

1.  $b_i = 2$ , es decir, solamente se llama a jugador (7 puntos)
2.  $x_i \leq 500$  (11 puntos)
3.  $\sum_{b_i=1} x_i \leq 10^6$ , es decir, la suma total de tiempo transcurrido es a lo sumo 1.000.000 (20 puntos)
4. Se realiza a lo sumo una llamada a tiempo, y se hace luego de todas las llamadas a jugador (15 puntos)
5. Se realiza a lo sumo una llamada a tiempo (22 puntos)
6. Sin más restricción (25 puntos)