

9

Problema 7

CHAVES LÓPEZ, ALBERTO (TAIS20)

ID envío	Usuario/a	Hora envío	Veredicto
55715	TAIS20	2021-09-21 10:52	AC
55709	TAIS20	2021-09-21 10:45	AC
55693	TAIS20	2021-09-21 10:36	WA
55679	TAIS20	2021-09-21 10:28	WA
55631	TAIS20	2021-09-21 10:05	WA

Fichero prob-07.cpp

*
 * Indicad el nombre completo y usuario del juez de quienes habéis hecho esta solución:
 * Estudiante 1:Alberto Chaves López TAIS20
 * Estudiante 2:Jose Otegui Marin
 *

Hemos creado un `struct` que contiene la informacion de la caja con su id y el tiempo que tiene que esperar en cada una de ellas, inicialmente inicializados los tiempos a 0 y el id asignado en orden.

Estas cajas estan ordenadas de menor a mayor con un `operator<` que reordena por el menor tiempo

de espera en cada caja y en caso de ser el mismo por el menor id.

Cada vez que entra un cliente se le suma el tiempo de la caja con menor tiempo, que en el caso

de estar las cajas vacias será cero, y en el resto será el del tiempo menor (`top()`).

Al estar ordenados de menor a mayor la respuesta siempre será la consulta `top()` ya que será el tiempo

menor acumulado de personas que han entrado en esa caja.

COSTE:

El coste de de insertar en una cola de prioridad es $O(\log n)$ por lo tanto al insertarse `nClientes` en la cola

el coste será de `nClientes <> O(N)`.

```
struct caja {
    int id;
    int tiempo;
};
```

```
bool operator <(caja a, caja b) {
    if (a.tiempo == b.tiempo)
        return a.id < b.id;
    else
        return a.tiempo < b.tiempo;
}
```

```
bool resuelveCaso() {
```

$O(\log n)$
 $O((\log N))$
 (clientes
 N cajas

```
// leer los datos de la entrada
int nCajas, nClientes;
std::cin >> nCajas;
std::cin >> nClientes;
if (nCajas==0 && nClientes==0)
    return false;

PriorityQueue<caja> colaCajas;
caja cajaVacía;
cajaVacía.id = 0;
cajaVacía.tiempo = 0;

for (int i = 0; i < nCajas; i++) {
    cajaVacía.id = i + 1;
    colaCajas.push(cajaVacía);
}
int tiempoCliente;
caja aux;
for (int i = 0; i < nClientes; i++) {
    cin >> tiempoCliente;
    aux.id = colaCajas.top().id;
    aux.tiempo = tiempoCliente + colaCajas.top().tiempo;
    colaCajas.pop();
    colaCajas.push(aux);
}
cout << colaCajas.top().id << "\n";

return true;
}
```

