

Problema 7

PAREDES SÁNCHEZ, MICHAEL STEVEN (TAIS74)

ID envío	Usuario/a	Hora envío	Veredicto
55637	TAIS74	2021-09-21 10:07	AC
55604	TAIS74	2021-09-21 09:57	AC
55587	TAIS74	2021-09-21 09:52	AC

Fichero Source.cpp

```
*
* Nombre y Apellidos: IVAN GONZALEZ TAIS50
* MICHAEL PAREDES TAIS74
*
```

Escribe aquí un comentario general sobre la solución, explicando cómo se resuelve el problema y cuál es el coste de la solución, en función del tamaño del problema.

-Tenemos una cola de prioridad, que nos indica los clientes que estan siendo atendidos en ese momento, y un vector de tiempo de clientes por atender.
Vamos sacando de la cola los clientes atendidos, y vamos añadiendo a su lista el siguiente cliente del vector en su caja y el tiempo acumulado de los anteriores clientes + su tiempo.

En realidad la cola es de cajas

-El coste seria $n \log(n)$, siendo n el numero de clientes. Metemos n clientes en una cola de prioridad, y su reordenacion de prioridad es $\log(n)$, por lo que $n * \log(n) = n \log(n)$

cajas

Hay dos tamaños

*N cajas
C clientes
 $O(C \log N)$*

```
bool resuelveCaso() {
    // leer los datos de la entrada
    int N, C;
    cin >> N >> C;
    if (N == 0 && C == 0)
        return false;

    vector<int> tiempo_cliente;
    PriorityQueue< cola_clientes> cajas;
    int aux;
    for (int i = 0; i < C; i++)
    {
        cin >> aux;
        tiempo_cliente.push_back(aux);
    }
    // resolver el caso posiblemente llamando a otras funciones
    int atendiendo = 0;

    if (C < N) cout << C + 1 << '\n';
```

```
else {
    for (int i = 0; i < N && i < C; i++)
    {
        cajas.push({ tiempo_cliente[i], i });
        atendiendo++;
    }
    cola_clientes aux;
    while (atendiendo < C) {
        aux = cajas.top();
        cajas.pop();
        cajas.push({ aux.tiempo + tiempo_cliente[atendiendo], aux.caja });
        atendiendo++;
    }
    cout << cajas.top().caja + 1 << '\n';
}

// escribir la solución

return true;
}
```

