Envío 75559 — 27/9 10:58

WA

TAIS69 | Íñigo Velasco Sánchez

main.cpp

```cpp
/*  *
 * Nombre y Apellidos: TAIS69 Íñigo Velasco Sánchez
 *
*/

/*
 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.
 Creo un struct Candidatura para albergar los datos de cada candidatura, con los votos de la entrada e inicializo a 0 los escaños.
 Además incluyo el índice para saber el orden de la entrada y así tenerlo en cuenta en el comparador y al escribir la salida.
 En el comparador, implemento las fórmulas que especifica el enunciado para hacer el cómputo de los votos según D'Hondt.
 El coste del bucle de lectura de los datos es de O(NlogN) para N candidaturas
 El coste del bucle while que resuelve el problema es de O(ElogN) siendo E el número de escaños a repartir y N el número de candidaturas en la cola.
*/

struct Candidatura {
    lli votos, escanios, indice;//indice para ver el orden de la entrada
};
bool operator<(Candidatura const& a, Candidatura const& b) {
    lli ca = a.votos / (1 + a.escanios);//coeficiente de candidatura a
    lli cb = b.votos / (1 + b.escanios);//coeficiente de candidatura b
    if (ca == cb) {
        if (a.votos == b.votos) return a.indice > b.indice;
        else return a.votos < b.votos;
    }
    return ca < cb;
}
bool resuelveCaso() {

    // leer los datos de la entrada
    lli C, N;
    cin >> C >> N;//C de candidaturas presentadas y el número N de escaños a repartir
    if (C == 0)
        return false;
    priority_queue<Candidatura> q;
    vector<Candidatura> R(C);
    for (int i = 0; i < C; i++) {
        lli votos;
        cin >> votos;
        q.push({ votos, 0, i });//O(logN)
    }
    // resolver el caso posiblemente llamando a otras funciones
    while (N > 0) {//mientras haya escaños para repartir
        Candidatura c = q.top(); q.pop();//top O(1), pop(logN)
        c.escanios += 1;//si ninguna candidatura tiene votos?
        N--;
```

```
        q.push(c);//coste O(logN)
    }
    // escribir la solución

    while (!q.empty()) {
        Candidatura c = q.top(); q.pop();
        R[c.indice] = c;
    }
    for (Candidatura& c : R) {
        cout << c.escanios << " ";
    }
    cout << "\n";
    return true;
}

```