

9'5

Problema 10

GONZÁLEZ RAMIRO, IVÁN (TAIS50)

ID envío	Usuario/a	Hora envío	Veredicto
56713	TAIS50	2021-09-28 10:00	AC
56693	TAIS50	2021-09-28 09:54	AC
56678	TAIS50	2021-09-28 09:47	WA
56676	TAIS50	2021-09-28 09:46	WA
56672	TAIS50	2021-09-28 09:45	CE
56642	TAIS50	2021-09-28 09:20	WA

Fichero Source.cpp

*
 * Indicad el nombre completo y usuario del juez de quienes habéis hecho esta solución:
 * Estudiante 1: IVÁN GONZÁLEZ RAMIRO TAIS 50
 * Estudiante 2: MICHAEL STEVEN PAREDES SÁNCHEZ TAIS 74
 *

Tengo un `struct` donde pongo cuantos musicos de un tipo hay con las partituras que tiene ese grupo y las personas por partitura.

Meto los musicos en un `priorityqueue` y los voy sacando y sumando una nueva partitura y recalculando el `porpartitura` y lo vuelvo a meter

asi hasta que me quedo sin partituras(el bucle `for` es de 0 a P-N ya que al meterlos inicialmente ya les doy una partitura)

Guardar los musicos en la `priorityqueue` tiene un coste de $\log m$ y como lo hago m veces tiene un coste de $m \log m$ siendo m la cantidad de musicos diferentes

luego repartir las partituras tiene un coste de $\log m$ y como lo hago $p-m$ veces siendo m los musicos diferentes y p las partituras

el coste resultante es de $m \log m + (p-m) \log m$ siendo m los musicos diferentes y p las partituras

```
struct musicos {
    double cantidad;
    double porpartitura;
    double partituras;
};

bool operator<(musicos const a, musicos const b) {
    return a.porpartitura < b.porpartitura || (a.porpartitura==b.porpartitura && a.cantidad<b.cantidad);
}
```

```
bool resuelveCaso() {
    // leemos la entrada
    int P, N;
    cin >> P >> N;
```

```

if (!cin)
    return false;
priority_queue<musicos> m;
int aux;

for (int i = 0; i < N; i++) {
    cin >> aux;
    musicos aux1 = { aux,aux,1};
    m.push(aux1);
}
musicos aux2;
for (int i = 0; i < P-N; i++) {
    aux2 = m.top();
    m.pop();
    aux2.partituras++;
    aux2.porpartitura = aux2.cantidad / aux2.partituras;
    m.push(aux2);
}
cout << ceil(m.top().porpartitura) << "\n";
// leer el resto del caso y resolverlo

return true;
}

```

