7

Problema 34

PAREDES SÁNCHEZ, MICHAEL STEVEN (TAIS74)

ID envio	Usuario/a	Hora envío	Veredicto
61299	TAIS74	2022-11-02 09:37	AC
61256	TAIS74	2022-11-02 09:18	AC

Fichero Source.cpp

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

El coste del algoritmo es 0(N log N), siendo N, el numero de partidos que se juega. El coste
    es el indicado debido a que
tenemos una priority_queue que ordena los puntos de los partidos de menor a mayor en los
    rivales, y de mayor a menor en
los Boston. Tras ello, en la funcion resultados, mandamos ambas pq y sumamos la diferencia
    entre los puntos de los boston
y los puntos de los rivales, si y solo si su resultado es mayor que 0. Los sacamos de la pq,
    y asi hasta tener las pq vacias
o si los resultados de los boston son menores a los de su rival.

Con esta funcion nos aseguramos que siempre haya la maxima diferencia positiva posible entre
    los resultados de los boston y sus rivales

int resultados(priority_queue<int, vector<int>, greater <int>> r, priority_queue<int> b) {
    int cont = 0, n = b.size();

    for (int i = 0; i < n; i++)
    {
        if (b.top() > r.top()) cont += (b.top() - r.top());
        else break;// esto es porque nunca más va a encontrar un resultado de los boston
    mayor a su rival
        b.pop();
        r.pop();
    }
    return cont;
}

bool resuelveCaso() {

    // leer los datos de la entrada
    int N, aux;
    cin >> N;
    if (N == 0)
        return false;
```

Falta demostración de la corrección de la estrategia voraz

```
    priority_queue<int, vector<int>, greater <int>> rivales;
    priority_queue<int> boston;
    for (int i = 0; i < N; i++)
    {
        cin >> aux;
        rivales.push(aux);
    }
    for (int i = 0; i < N; i++)
    {
        cin >> aux;
        boston.push(aux);
    }
    // resolver el caso posiblemente llamando a otras funciones
    cout << resultados(rivales, boston) << '\n';
    // escribir la solución

    return true;
}
```