

Ejercicios Tema 7 sobre el TAD SetOfInts

Se pide lo siguiente:

1. Implementa la relación de equivalencia del TAD `SetOfInts` mediante el método `bool operator==(const SetOfInts<T>& set)` para las tres implementaciones del TAD vistas en clase y disponibles en el CV. Escribe una función que permita hacer pruebas repetidamente con conjuntos introducidos por el usuario.
2. Añade una nueva operación al TAD `SetOfInts3` que compruebe si un conjunto es un subconjunto de otro mediante la redefinición del operador de orden `<`, es decir, mediante el método `bool operator<(const SetOfInts3<T>& set)`. Escribe una función que permita hacer pruebas repetidamente con conjuntos introducidos por el usuario.
3. Escribe una función (fuera del TAD) que dada una serie de n números enteros (en un array o vector), encuentre los k números menores distintos, devolviendo el resultado en un conjunto `SetOfInts3` (como parámetro por referencia). Se extenderá la clase `SetOfInts3` añadiendo dos métodos, uno para consultar el mayor elemento del conjunto (`getMax`) y otro para eliminarlo (`removeMax`), y por tanto el algoritmo diseñado para la función deberá usarlos. Indica en un comentario la complejidad de los algoritmos implementados.

Instrucciones de entrega:

Debes subir al juez online los ficheros `Error.h`, `SetOfInts3.cpp` y `SetOfInts3.h`, éstos incluyendo las nuevas operaciones, y el `Main.cpp` con la función del tercer apartado y la función `main`. El corrector automático del juez solo comprobará la validez de la función correspondiente al tercer apartado (ver abajo la descripción de E/S esperadas).

Entrada

La entrada que espera el corrector automático consta de una serie de casos de prueba y acabará cuando se introduzca un 0. Cada caso de prueba está formado por dos líneas. La primera contendrá el valor $0 < k \leq 50$, que será menor o igual que el número de elementos (distintos) de la serie. La segunda línea contendrá los elementos de la serie (posiblemente con repeticiones) todos positivos. El fin de la serie vendrá indicado con un -1.

Salida

Para cada caso de prueba se escribirá una línea con los k elementos menores de la serie, sin repeticiones y ordenados de menor a mayor.

Entrada de ejemplo

```
2
7 3 2 3 8 7 -1
3
1 8 3 14 5 -1
3
6 5 4 3 2 1 1 1 -1
0
```

Salida de ejemplo

```
2 3
1 3 5
1 2 3
```