

Los k elementos mayores usando el TAD Set

Dada una serie de elementos ordenables, queremos encontrar los k elementos distintos mayores. En concreto, se pide lo siguiente:

1. Extiende la clase `Set` vista en clase (y disponible en el CV) añadiendo los métodos `getMax`, `removeMax`, `getMin` y `removeMin`, que obtienen/borran el máximo y el mínimo elemento del conjunto respectivamente. Indica la complejidad de los métodos implementados (como comentario junto a ellos en el código).
2. Escribe una función (fuera del TAD) que dada una serie de elementos ordenables, encuentre los k elementos mayores distintos, devolviendo el resultado en un `Set` (como parámetro por referencia). El algoritmo diseñado no debe en ningún caso ordenar toda la serie y debería usar los nuevos métodos `getMin` y `removeMin`.
3. Justifica la complejidad del algoritmo implementado. Se espera que ésta dependa también de k y no solo de la longitud de la serie de entrada.
4. Indica cómo se podría conseguir que la operación `removeMin` tuviese complejidad constante. ¿Sería posible conseguir que tanto `removeMin` como `removeMax` fuesen constantes sin empeorar la complejidad de ninguna otra operación? En tal caso indica cómo. Escribe tus respuestas como un comentario junto a la función `main`.

Instrucciones de entrega:

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

Entrada

Cada caso de prueba está formado por tres líneas. La primera contendrá el carácter N si los elementos de la serie son números, o el carácter P si los elementos son palabras. La segunda línea contendrá el valor $k > 0$, que será menor o igual que el número de elementos (distintos) de la serie. La tercera línea contendrá los elementos de la serie (posiblemente con repeticiones). Si son números estarán en el rango $[0..10^9]$, y el fin de la serie vendrá indicado con un `-1`. Si son palabras, estarán formadas por no más de 30 caracteres de la 'a' a la 'z', y el final de la serie estará indicado con la palabra `FIN`. El programa procesará casos de prueba repetidamente.

Salida

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

Entrada de ejemplo

```
N
3
1 8 3 14 5 -1
P
2
maria luis marta juan alberto FIN
N
3
1 2 3 4 5 6 6 6 -1
```

Salida de ejemplo

```
5 8 14
maria marta
4 5 6
```