

### Tema: Clasificación. MSD Radix Sort

En muchas aplicaciones se aprovecha el hecho de que las claves de los registros a clasificar pueden considerarse como números de algún intervalo finito. Los métodos de ordenación que aprovechan estas propiedades numéricas de las claves se llaman ordenaciones por residuos. Estos métodos no comparan las claves, procesan y comparan fragmentos de ellas.

### Clasificación por residuos (Radix Sort)

La ordenación por residuos podría caracterizarse como una ordenación de “utilidad especial”, porque su viabilidad depende de propiedades especiales de las claves, en contraste con la “utilidad general” de algoritmos tales como el de ordenación rápida, que se utilizan mucho más porque se adaptan a una gran variedad de aplicaciones.

Este método de clasificación era el utilizado por antiguas máquinas encargadas de ordenar las tarjetas perforadas de las computadoras: un lote de tarjetas se procesaba 80 veces, una por cada columna, procediendo de derecha a izquierda. Basado en estas ideas, Harold H. Seward (MIT) inventó un primer algoritmo para Radix Sort en el año 1954.

En su forma más simple este método se basa en la representación posicional de los dígitos de los números enteros que se procesan como claves. La **clave más grande** de dos claves enteras **de igual longitud** se determina de la siguiente forma: se comienza con el dígito más significativo y se avanza por los dígitos menos significativos mientras coincidan los dígitos correspondientes en los dos números enteros. El número con el dígito más grande en la primera posición en la cual los dígitos de los dos números no coincidan es el mayor de los dos. Por supuesto, si coinciden en todos sus dígitos los dos números son iguales.

Se pueden escribir dos algoritmos con esta idea, el primero permite ordenar claves numéricas comenzando por el dígito menos significativo (LSD) y el segundo por el dígito más significativo (MSD). En ambos casos si se utiliza base decimal, los números se dividen en diez grupos basándose en cada uno de los dígitos considerados. Para ordenar datos almacenados en una fila con esta idea se recorre la fila, ordenando por cada dígito de las claves. En el caso de utilizar MSD, se comienza por el dígito más significativo. En cada paso cada clave se coloca en una de diez filas, dependiendo del valor del dígito que se procesa en ese momento. Después se ordenan cada una de las 10 filas de forma recursiva, utilizando el siguiente dígito. Finalmente se vuelve a armar la fila original concatenando las 10 filas, comenzando por la del dígito 0 y siguiendo en orden hasta el 9. Cuando este paso se ha realizado para cada dígito, comenzando por el más significativo y terminando con el menos significativo la fila está ordenada.

- 1) Escriba Ud. un algoritmo recursivo que ordene datos numéricos almacenados en una fila con el método de residuos MSD, usando las operaciones ADT FILA y considerando  $n$  claves enteras de  $k$  dígitos cada una.
- 2) Codifique el algoritmo del punto 1) utilizando una FILA de enteros implementada con punteros. Pruebe su programa con distintas cantidades de datos y diferentes tipos de filas (ordenadas, aleatorias, en orden inverso).
- 3) Usando como base el algoritmo que escribió en el punto 1) conteste las siguientes preguntas que le permitirán resumir las características que tiene el método de ordenación por residuos.
  - ¿Cuál es el tiempo de ejecución del algoritmo para ordenar  $n$  claves de longitud  $k$ . ¿Qué ocurre cuando  $k \ll n$ ? y cuando  $k \approx n$ ?
  - ¿Necesita memoria extra?
  - ¿Es estable? ¿Por qué?
  - ¿Es sensible al orden inicial de los datos? ¿Por qué?
  - ¿Cuáles serían los cambios en su algoritmo para poder usar claves numéricas en otra base de numeración (por ej. 2, 8 o hexadecimal)?
  - ¿Se podría adaptar su algoritmo a claves alfabéticas? ¿En caso afirmativo, cómo?