G4. Maria Montalvo, Ricardo López, Sven Wallin i Victor Pla

2.14. Tenim un taula T amb n claus (no necessàriament numèriques) que pertanyen a un conjunt totalment ordenat. Doneu un algorisme $O(n + k \log k)$ per a ordenar els k elements a T que són els més petits d'entre els més grans que la mediana de les claus a T.

Idea principal

Para encontrar los elementos más grandes que la mediana:

- Es un conjunto totalmente ordenado.
- Tendremos que encontrar la mediana.
- Una vez tengamos el array con los elementos más grandes que la mediana, encontrar el elemento k-ésimo.

Pseudocodigo

Parte 1 : Encontrar la mediana

a) Problema de la selección de mediana:

1. Dividimos los n elementos del array⌊n/5⌋ en grupos de 5 elementos cada uno. Habrá un array con *n* mod 5 elementos.
2. Encontraremos la mediana de cada grupo (con 25 comparaciones o con el algoritmo de inserción).
3. Utilizamos select recursivamente para encontrar la mediana de las medianas encontradas en el paso 2.
4. Particionamos el array por la mediana de medianas x utilizando la versión modificada de partition. Sea k uno más que el número de elementos de la parte baja de la partición, tal que x es el k-ésimo elemento y hay n-k elementos en la parte alta de la partición.
5. Si i ==k devolvemos x. Si no, usamos el algoritmo recursivamente para encontrar el elemento en la parte baja si i<k o en la parte alta si i>k.

Select(A, i)
Divide A into $m = \lceil n/5 \rceil$ groups, all but at most one with 5 elements
$X[j]$ = median of group j, $j = 1, \ldots, m$
x = **Select**$(X, \lfloor (m+1)/2 \rfloor)$ i.e. the median of X
Let k be the rank of x in A
if $i = k$ **then**
  return x
else
  L = the elements of A smaller than x
  R = the elements of A bigger than x
  if $i < k$ **then**
    return **Select**(L, i)
  else
    return **Select**$(R, i - k)$

Aprovechamos que el algoritmo nos devuelve la parte derecha de la mediana del array (elementos más grandes que la mediana).

Esta parte de la solución tiene coste lineal.

b) Problema de selección del késimo de los elementos más grandes que la mediana (aplicando el algoritmo anterior). Nos quedaremos con un subarray de k elementos más grandes que la mediana.

Esta parte de la solución tiene coste lineal.

c) Problema ordenación k elementos:
    1. Utilizar merge sort sobre los k elementos.

Esta parte de la solución tiene coste k*log k

Correctitud

Supongamos S la solución para un problema cualquiera.
Supongamos u el elemento máximo de la solución S.
Si u no formase parte de la solución:

- Tenemos una solución con k-1 elementos, u podría ser añadido en la solución.
- Tenemos un elemento fuera de la solución más pequeño que u, cosa que no puede ser ya que hemos dicho que S era solución y que u era el elemento máximo de la solución (el elemento k)

Tiempo de ejecución

- Punto a: O(n) para seleccionar la mediana
- Punto b: O(n) para seleccionar el k-ésimk
- Punto c: O(k*logk)

Justificación

Parte lineal de selección.
Parte de ordenación con coste k*log k