

Auxiliar 7 - Dominios Discretos

CC4102 - Diseño y Análisis de Algoritmos

Profesor: Gonzalo Navarro Auxiliar: Jorge Bahamonde

2 de Noviembre del 2015

1. Se desea ordenar un arreglo de llaves cuyos valores pueden ser 0 o 1. Algunas características deseables son las siguientes:

- (a) El algoritmo toma tiempo $O(n)$.
- (b) El algoritmo es estable.
- (c) El algoritmo ordena de forma *in-place*: es decir, el espacio adicional utilizado para ordenar es constante.

Diseñe algoritmos que cumplan (a) y (b); (a) y (c); (b) y (c).

2. Ordene n números en el rango $[0, n^2 - 1]$ en tiempo $O(n)$. Generalice su resultado para dominios de la forma $[0, n^k - 1]$, para k constante.
3. Sea B una secuencia de bits de largo n . Supongamos que acceder a un bit $B[i]$ Se define $\text{RANK}_b(B, i)$ como el número de bits con valor b en $B[1, i]$. En particular:

$$\text{RANK}_1(B, i) = \sum_{0 < j \leq i} B_j, \quad 1 \leq i \leq n$$

Se define, además, $\text{SELECT}_b(B, i)$ como la posición de la i -ésima repetición del valor b en B .

- (a) Sea $A[1, t]$ un arreglo de t enteros no negativos que suman n . Muestre cómo realizar las siguientes consultas usando RANK y SELECT :
 - $\text{SUM}(r)$: el valor de $\sum_{j=1}^r A[j]$.
 - $\text{SEARCH}(s)$: el mínimo valor de r para el cual $\sum_{j=1}^r A[j] \geq s$.
 - (b) Construya una estructura que permita calcular $\text{RANK}(B, i)$ en tiempo constante, utilizando $2n + o(n)$ **bits** de espacio.
 - (c) Resuelva el mismo problema, esta vez utilizando $o(n)$ bits de espacio.
 - (d) **Propuesto** (fácil): Construya una estructura que permita calcular $\text{SELECT}(B, i)$ en tiempo $\Theta(\log \log n)$ que utilice $o(n)$ bits de espacio.
 - (e) **Propuesto** (no-fácil): Construya una estructura que permita calcular $\text{SELECT}(B, i)$ en tiempo constante, usando $o(n)$ bits de espacio.
4. Describa un algoritmo que, dados n enteros en $[0, \dots, k]$, preprocese su entrada y responda cuántos de estos enteros se encuentran en el rango $[a, \dots, b]$ en tiempo constante. Su algoritmo debería tomar tiempo $\Theta(n + k)$ en el preprocesamiento.