

CC4102 - Diseño y Análisis de Algoritmos

Auxiliar 6

Prof. Gonzalo Navarro; Aux. Mauricio Quezada

7 de Diciembre, 2011

1 Dominios finitos y discretos: Select y Rank en bitmaps

Sea $B[1, n]$ un arreglo de n bits, y supongamos que acceder a un bit $B[i]$ toma tiempo constante. Definamos las siguientes operaciones:

- $rank_b(B, i)$: el número de veces que se repite el bit b en $B[1, i]$
- $select_b(B, i)$: la posición de la i -ésima repetición de b en B

1.1 Ejercicios

1. Muestre cómo construir una estructura de hashing perfecto estática (es decir, donde la probabilidad de colisión es despreciable) utilizando la operación $rank$, asumiendo que tiene un universo de n llaves y desea almacenar t llaves de r bits.
2. Sea $A[1, t]$ un arreglo de t enteros no negativos que suman n . Muestre cómo realizar las siguientes consultas usando $rank$ y $select$:
 - $Sum(r)$: el valor de $\sum_{j=1}^r A[j]$
 - $Search(s)$: el mínimo valor de r para el cual $\sum_{j=1}^r A[j] > s$
3. Muestre cómo realizar la operación $rank$ en tiempo constante utilizando $O(n \log n)$ bits de espacio. ¿Se pueden usar sólo $O(n)$ bits? Muestre cómo lograr $3n + o(n)$ bits. ¿Y $n + o(n)$?

2 Repaso para el control

2.1 P1 C1 2009/1

Dado un arreglo desordenado de elementos que sólo pueden compararse por $<, =, >$, se desean obtener los k menores elementos del arreglo, en orden creciente.

1. Diseñe un algoritmo que lo resuelva en tiempo $O(n + k \log k)$
2. Demuestre que este algoritmo es óptimo
3. Considere la variante *online* del problema, donde no se conoce k , sino que se pide el siguiente mínimo hasta k sin saber cuándo se detendrá. Diseñe un algoritmo óptimo para este problema.

2.2 P3 Examen 2010/2

Considere el problema de determinar cuánto tiempo se cuece un huevo en agua hirviendo, dado el acceso a una gran cantidad de huevos de tamaños similares. Una vez que se escoge un huevo, a priori se elige el tiempo (en minutos) en que se cocinará y sólo después de ese tiempo se puede abrir el huevo para ver su estado: muy cocido, a punto, o muy crudo. Una vez abierto, no se puede volver a cocinar.

1. Describa un experimento para calcular el tiempo t en minutos de cocción ideal (por ejemplo, *a la copa*) de un huevo en agua hirviendo, con una precisión de 1 minuto, tal que la cantidad de huevos usados sea $o(t)$, y justifique su análisis.
2. Considere el caso donde los huevos llegan uno a la vez y son de tamaños diferentes, donde v_i es el volumen del i -ésimo huevo. Asuma que el tiempo t ideal para un huevo depende linealmente de su volumen v , con un factor constante τ desconocido. Modifique su diseño experimental de forma de calcular τ y luego $t(v)$ para un huevo dado, con una precisión de 1 minuto. ¿Cuál es la complejidad de este algoritmo en el peor caso para τ fijo? Y comparado con la versión *offline*?

2.3 Para ejercitar

1. Notación asintótica: verdadero o falso:
 - (a) $n^2 + 10000n \in O(n^2)$
 - (b) $n^2 - n \in \Omega(n^2)$
 - (c) $2^{n+1} \in O(2^n)$
 - (d) $4^{\lg n} \in \Theta(n^2)$
 - (e) $2^{2n} \in O(2^n)$
 - (f) $\sqrt{n} \log n \log \log n \in o(n)$
 - (g) $n + k \log n \in \Theta(n + k \log k)$, con $0 \leq k \leq n$
2. Muestre la cota superior de Mergesort Multiario en términos de accesos a la memoria externa, considerando una memoria de tamaño M , páginas de tamaño B y el arreglo a ordenar de tamaño N .
3. De un código de Huffman óptimo para el siguiente conjunto de símbolos con sus frecuencias: $a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21$. ¿Puede generalizar para el caso de los n primeros números de Fibonacci?.
4. Una secuencia de n operaciones se realizan sobre una estructura de datos. La operación i -ésima cuesta i , si i es una potencia de 2, y cuesta 1 sino. Use el análisis agregado para determinar el costo amortizado por operación.
5. Los controles en U-Cursos.