Universidad de Chile
Departamento de Ciencias de la Computación
CC4102 - Diseño y Análisis de Algoritmos

Auxiliar 6 - "Universos Discretos y Finitos"

Profesores: Pablo Barceló
Gonzalo Navarro
Auxiliar: Dustin Cobas

P1. Ordenando Strings

Dados n strings $s_1, s_2, \ldots, s_n \in \Sigma^*$, con $\sigma = |\Sigma| \in \mathcal{O}(n)$, queremos ordenarlos lexicográficamente. Además, el largo total de los strings es $N = \sum_{i=1}^{n} |s_i|$.

Diseñe algoritmos de tiempo $\mathcal{O}(N)$ en los casos:

a) Los strings tienen el mismo largo m, por lo que $N = nm$.

b) Los strings tienen largo variable $m_i \geq 1$.

P2. Rank

Sea B una secuencia de bits de largo n. Se define $\texttt{rank}(B, i)$ como la cantidad de bits 1 en $B[1, i]$, es decir:

$$\texttt{rank}(B, i) = \sum_{0 < j \leq i} B[j],\ 1 \leq i \leq n$$

Construya una estructura que permita calcular $\texttt{rank}(B, i)$ en tiempo constante usando:

a) $2n + o(n)$ **bits** de espacio.

b) $o(n)$ **bits** de espacio.

Soluciones

P1. Ordenando Strings

Apuntes del Curso, Sección 4.1.4.

P2. Rank

a) Una primera idea consiste en tener un arreglo $A[1,n]$, tal que $A[i] = \texttt{rank}(B,i)$. Como `rank` puede llegar hasta n, necesitaríamos $\mathcal{O}(\log n)$ bits para representar cada valor de A. En total el arreglo A requeriría $\mathcal{O}(n \log n)$ bits.

Podemos dividir B en partes de tamaño $\frac{\lceil \log n \rceil}{2}$ y guardar el `rank` de los últimos elementos de cada una de las partes. Esto toma $\frac{n}{\frac{\lceil \log n \rceil}{2}} \log n = \frac{2n}{\lceil \log n \rceil} \log n \leq 2n$ bits de espacio.

Ahora que tenemos el `rank` cada $\frac{\lceil \log n \rceil}{2}$ solo debemos identificar el `rank` más cercano y luego calcular el resto en tiempo $\mathcal{O}\left(\frac{\lceil \log n \rceil}{2}\right)$.

Para reducir la complejidad a $\mathcal{O}(1)$, precalcularemos todos los `rank` de todas las secuencias de largo $\frac{\lceil \log n \rceil}{2}$. Es decir, tendremos en una tabla $T[w,r] = \texttt{rank}(w,r)$ que ocupará espacio $2^{\frac{\lceil \log n \rceil}{2}} \cdot \frac{\lceil \log n \rceil}{2} \cdot \log \frac{\lceil \log n \rceil}{2} = \mathcal{O}(\sqrt{n}) \cdot \mathcal{O}(\log n) \cdot \mathcal{O}(\log \log n) \in o(n)$ bits.

b) El problema con la solución anterior es que el arreglo que guarda los `rank` de las partes ocupa $\mathcal{O}(n)$ bits de espacio.

Podemos aumentar el tamaño de las partes a $\frac{\log^2 n}{2}$, de este modo solo necesitamos $\frac{n}{\frac{\log^2 n}{2}} \log n = \frac{2n}{\log n} = o(n)$ bits de espacio.

Lamentablemente, ya no podemos guardar todos los `rank` de todas las secuencias de tamaño $\frac{\log^2 n}{2}$ en espacio $o(n)$ bits.

Dividimos cada parte en $\log n$ subpartes de tamaño $\frac{\log n}{2}$ cada una y guardamos el `rank` del final de cada una, pero relativo a la parte en la cual se encuentra esa subparte. De este modo necesitamos $\log n \cdot \frac{n}{\frac{\log^2 n}{2}} \cdot \log\left(\frac{\log^2 n}{2}\right) \leq \frac{4n \log \log n}{\log n} = o(n)$ bits.