

CC4102 - Control 3

Prof. Gonzalo Navarro

17 de Diciembre de 2020

P1 (2.0 pt)

Un índice invertido para un texto largo n (medido en palabras) palabras almacena, para cada palabra distinta, la lista creciente de las posiciones (enteros en $[1, n]$) donde la palabra aparece. Se desea crear estructuras de datos apropiadas para buscar frases de dos palabras, es decir, dadas dos palabras X, Y encontrar todos los i donde X aparece en la posición i e Y aparece en la posición $i + 1$. Para los problemas que siguen, puede asumir que tiene el tiempo que desee para crear las estructuras, las que deben ocupar espacio $O(n)$.

1. (0.5pt) Muestre cómo resolver el problema en tiempo esperado $O(\min(n_X, n_Y))$, donde n_X y n_Y son las cantidades de veces que X e Y aparecen en el texto, respectivamente.
2. (1.5pt) Muestre cómo resolver el problema en tiempo de peor caso $O(A_{X,Y} \log \log n)$, donde $A_{X,Y} \leq \min(n_X, n_Y)$ se define como la cantidad de veces que, si dejamos sólo las palabras X e Y en el texto, una Y sigue a una X o viceversa (por ejemplo, si todas las X vienen antes de todas las Y , entonces $A_{X,Y} = 1$).

P2 (2.0 pt)

Se busca particionar un string $S[1..n]$ en bloques de largo 1 ó 2 con propiedades de “consistencia local”, es decir que el mismo substring en dos partes de S se particionen de la misma forma. Para ello, se decide aleatoriamente que cada símbolo del alfabeto es “izquierdo” o “derecho” con probabilidad $1/2$. Luego, cada substring ab en S , donde a sea izquierdo y b sea derecho, se une en un solo bloque. Los demás símbolos se dejan como bloques de largo 1, sin aparear. Por ejemplo, si $S[1..16] = abcdcbadbcbababab$, y decidimos que a y c son izquierdos mientras que b y d son derechos, la partición será $S = ab|cd|cb|ad|b|c|ab|ab|ab$.

1. (0.7pt) Demuestre que, si S no contiene pares de caracteres repetidos (como aa), el número esperado de bloques en S es a lo más $(3/4)n + 1/4$.
2. (0.7pt) Diseñe un algoritmo tipo Las Vegas que obtenga una partición con a lo sumo $(4/5)n$ bloques, y calcule su tiempo esperado. Recuerde la desigualdad de Markov, $Pr(X \geq a) \leq E(X)/a$ para una variable aleatoria X .
3. (0.6pt) Usando el punto anterior, considere generar una partición jerárquica de S , mediante tomar los bloques producidos como nuevos símbolos que forman un nuevo string S' , repetir el proceso sobre S' , y continuar hasta que quede un solo símbolo. Calcule el tiempo esperado total de esta construcción.

P3 (2.0 pt)

Considere dos árboles aleatorizados T_1 y T_2 , cuyos elementos son todos distintos. Se desea unir los árboles en un árbol T que sea un árbol aleatorizado correcto para la unión de los dos elementos (esto significa que la probabilidad de que T tenga una determinada forma debe ser la misma que si hubiéramos insertado los elementos de T_1 y T_2 en orden aleatorio uniforme en un árbol inicialmente vacío).

Diseñe un algoritmo eficiente para unir los dos árboles y muestre que T resulta ser un árbol aleatorizado correcto. No es aceptable recolectar los elementos e insertarlos en un árbol vacío, debe ser algo significativamente más eficiente.