UNIVERSIDAD DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

Matemáticas Discretas para la Computación - CC3101
Control 2 - Semestre Primavera 2014

1. Sea π una permutación de $\{1, \dots, n\}$. Decimos que π es un *desordenamiento* si $\pi(i) \neq i$, para todo $1 \leq i \leq n$.

   a) (4 pts) Demuestre que si D_n denota el número de desordenamientos sobre $\{1, \dots, n\}$, entonces $D_n = (n-1)(D_{n-1}+D_{n-2})$. (*Hint:* El factor $n-1$ aparece debido a que hay $n-1$ posibilidades de elegir $\pi(1)$ en un desordenamiento π de $\{1, \dots, n\}$).

   b) (2 pts) Utilizando lo anterior, demuestre que $D_n = nD_{n-1} + (-1)^n$, para todo $n \geq 2$.

   Solución: Considere los desordenamientos $\pi_k : \{1, \dots, n\}$ tal que $\pi(1) = k$, para $1 < k \leq n$. Los desordenamientos de la forma π_k son de dos tipos: (1) Aquellos tales que $\pi_k \neq 1$, y (2) aquellos tales que $\pi_k = 1$. Aquellos que son del primero tipo siempre se pueden obtener desde un desordenamiento π' de $\{2, \dots, n\}$ reemplazando a k por 1. Por tanto, existen D_{n-1} desordenamientos de este tipo. Aquellos que son del segundo tipo están en relación 1-1 con los desordenamientos del conjunto $\{2, \dots, k-1, k+1, \dots, n\}$. Por tanto, hay D_{n-2} de ellos. La cantidad de desordenamientos es, por ende, $(n-1)(D_{n-1} + D_{n-2})$.

   Para la parte (b) utilizamos inducción. Claramente esto se cumple para $n = 2$. Considere entonces D_n. Por parte (a), $D_n = (n-1)(D_{n-1} + D_{n-2})$. Por tanto, $D_n = nD_{n-1} + (n-1)D_{n-2} - D_{n-1}$. Pero por HI se tiene que $(n-1)D_{n-2} - D_{n-1} = -(-1)^{n-1} = (-1)^n$.

2. Considere la siguiente estrategia para buscar un elemento x en una lista A no ordenada: Elija un índice i al azar. Si el i-ésimo elemento de A es x, entonces terminamos. De otra forma, seguimos la búsqueda eligiendo un nuevo índice j al azar hasta que encontramos x. Note que en cada paso elegimos sobre el conjunto completo de índices, por lo que podríamos examinar un elemento más de una vez.

   a) (4 pts) Asuma que existe exactamente un índice i tal que la i-esima posición de A contiene a x. ¿Cuál es el número esperado de pasos que la estrategia realizará hasta encontrar x?

   b) (2 pts) ¿Qué sucede si hay $k \geq 1$ índices i que contienen a x?

   Hint: Recuerde que $\sum_{k=1}^{\infty} kx^k = \frac{x}{(1-x)^2}$ para $|x| < 1$.

   Solución: Consideremos (a) primero. Sea X el número de comparaciones realizadas por el algoritmo. Estamos entonces interesados en conocer $E(X)$. Por definición, $E(X) = \sum_{i\geq 1}^{\infty} i \cdot Pr(X = i)$. Además, se puede observar que $Pr(X = i) = (n - 1/n)^{i-1} \cdot 1/n$. Por tanto,

$$E(X) = \frac{\sum_{i\geq 1}^{\infty} i(n-1/n)^{i-1}}{n} = \frac{\sum_{i\geq 1}^{\infty} i(n-1/n)^{i}}{n-1}.$$

Claramente, $n - 1/n < n$, y por tanto, $E(X) = n$.

Para la parte (b) utilizamos el mismo razonamiento, solo que en este caso $E(X) = \frac{k}{n} \cdot \sum_{i \geq 1}^{\infty} i(n - k/n)^{i-1}$. Por tanto, $E(X) = n/k$.

3. Sea Σ un alfabeto finito y w un string (palabra) sobre Σ. Una palabra w' es una *subsecuencia* de w, si (a) $w' = w$, o (b) w' se puede obtener desde w borrando algunas de sus letras. Formalmente, si $w = a_1 a_2 \dots a_n$ ($n \geq 0$), donde cada a_i es una letra en Σ, entonces w' es subsecuencia de w si (a) w' es la palabra vacía, o (b) existen índices $1 \leq i_1 < \dots < i_k \leq n$ tal que $w' = a_{i_1} \dots a_{i_k}$. Por ejemplo, la palabra TALADRO tiene como subsecuencias a las palabras ARO y TAO (entre otras).

   Dadas dos palabras w_1 y w_2 sobre Σ, denotamos por $LCS(w_1, w_2)$ el largo de la palabra más larga que es subsecuencia de ambas w_1 y w_2. Por ejemplo, si $w_1 =$ AGCAT y $w_2 =$ GAT, entonces $LCS(w_1, w_2) = 2$, ya que las palabras más largas que son subsecuencias tanto de w_1 como de w_2 son GA, GT y AT.

   Asuma que (a) $w_1 = a_1 \dots a_n$ y $w_2 = b_1 \dots b_m$, donde los a_i's y los b_j's son letras en Σ, que (b) w'_1 es la palabra que se obtiene de w_1 al sacar la última letra a_n, y que (c) w'_2 es la palabra que se obtiene de w_2 al sacar la última letra b_m. Demuestre lo siguiente utilizando inducción (fuerte, estructural, o la que usted desee):

$$LCS(w_1, w_2) \;=\; \begin{cases} 0, & \text{si } w_1 \text{ o } w_2 \text{ es la palabra vacía,} \\ LCS(w'_1, w'_2) + 1, & \text{si } a_n = b_m, \\ \text{máx}\,\{LCS(w'_1, w_2), LCS(w_1, w'_2)\}, & \text{si } a_n \neq b_m. \end{cases}$$

   Solución: Lo demostraremos mediante inducción sobre el par (n, m) utilizando el orden lexicográfico en $\mathbb{N} \times \mathbb{N}$. Los casos base son todos aquellos que cumplen $n = 0$ o $m = 0$. Entonces alguna de las dos palabras es vacía y claramente $LCS(w_1, w_2) = 0$.

   Considere ahora el par $(n + 1, m + 1)$, para $n, m \geq 0$. Por tanto, ni w_1 ni w_2 es la palabra vacía. Considere w, una subsecuencia común a w_1 y w_2 de largo máximo. Entonces hay dos posibilidades: (a) $a_n = b_m$, o (b) $a_n \neq b_m$. En el primer caso se debe cumplir que $w = w' a_n$, donde w' es una subsecuencia común más larga de w'_1 y w'_2. Por tanto, por HI se cumple que $LCS(w_1, w_2) = LCS(w'_1, w'_2) + 1$. En el segundo caso, se debe cumplir que w es una subsecuencia común más larga de w'_1 y w_2, o una subsecuencia común más larga de w_1 y w'_2. Por tanto, por HI se cumple que $LCS(w_1, w_2) = \text{máx}\,\{LCS(w'_1, w_2), LCS(w_1, w'_2)\}$.