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

Matemáticas Discretas para la Computación - CC3101
Control 2 - Semestre Primavera 2017

1. Un *árbol binomial* B_k se define inductivamente de la siguiente forma:

   a) B_0 es un árbol de un solo nodo.

   b) B_{k+1} se obtiene a partir de dos árboles B_k, tomando uno de ellos y agregándolo como último hijo de la raíz del otro (el auxiliar hará un dibujo).

   Un *bosque binomial* es un conjunto de árboles binomiales donde no se repite ningún B_k.

   Demuestre las siguientes propiedades usando inducción:

   a) B_k tiene 2^k nodos y altura k (un nodo individual tiene altura 0).

   b) La cantidad de nodos a profundidad i en B_k es $\binom{k}{i}$ (la raíz está a profundidad 0). Hint: recuerde la identidad $\binom{k+1}{i} = \binom{k}{i} + \binom{k}{i-1}$.

   c) Para cada $n \in \mathbb{N}$, existe exactamente un bosque binomial con n nodos en total. *Hint:* considere hacer inducción sobre k, donde 2^k es la mayor potencia de 2 menor o igual a n.

   Solución:

   a) Nodos: Caso base $|B_0| = 1$ y caso inductivo $|B_{k+1}| = 2|B_k| = 2 \cdot 2^k = 2^{k+1}$. Altura: Caso base $h(B_0) = 0$ y caso inductivo $h(B_{k+1}) = 1 + h(B_k) = k + 1$.

   b) Inducción sobre k. Caso base $k = 0$ y entonces $i = 0$, hay $\binom{0}{0} = 1$ nodo a profundidad 0 (es decir, una sola raíz). Caso inductivo, los nodos a profundidad i en B_{k+1} son los que están a profundidad i en el primer B_k más los que están a profundidad $i-1$ en el otro B_k que se cuelga del primero, entonces por HI tenemos $\binom{k}{i} + \binom{k}{i-1} = \binom{k+1}{i}$ nodos a profundidad i.

   c) Inducción sobre k como se sugiere. Si $k = 0$ entonces $n = 1$ y hay un único bosque binomial de 1 nodo: el que contiene sólo a B_0, pues si no tendría más de 1 nodo. Si $k > 0$, entonces $2^k \leq n < 2^{k+1}$. Para sumar n nodos debemos incluir B_k, pues con los k menores no sumamos el 2^k que requerimos tener, y con B_{k+1} nos pasamos de n. Si incluimos B_k, debemos completarlo con un bosque binomial para $n' = n - 2^k$, que como cumple $n' < 2^k$, no usará B_k. El k' que le corresponderá (máximo tal que $2^{k'} \leq n'$) es $k' < k$, por lo que por HI tenemos un único bosque binomial que lo forma. Entonces tenemos la tesis inductiva.

2. Con la definición de cadenas (Σ^*) por inducción estructural vista en clase, defina las funciones:

   a) $substring(w, w')$, que entregue 1 si w' es un substring (subcadena contigua) de w, y 0 si no.

   b) $subsecuencia(w, w')$, que entregue 1 si w' es una subsecuencia (no necesariamente contigua) de w, y 0 si no.

   Solución:

   a) Inducción sobre ambas cadenas.

- $substring(w, \varepsilon) = 1$.
- $substring(\varepsilon, w'b) = 0$.
- $substring(wa, w'b) = \max(\mathit{suffix}(wa, w'b), substring(w, w'b))$.

Usamos la función auxiliar $\mathit{suffix}(w, w')$, que también definimos recursivamente:

- $\mathit{suffix}(w, \varepsilon) = 1$.
- $\mathit{suffix}(\varepsilon, w'b) = 0$.
- $\mathit{suffix}(wa, w'a) = \mathit{suffix}(w, w')$.
- Si $a \neq b$, $\mathit{suffix}(wa, w'b) = 0$.

b) Inducción sobre ambas cadenas.

- $subsecuencia(w, \varepsilon) = 1$.
- $subsecuencia(\varepsilon, w'b) = 0$.
- $subsecuencia(wa, w'a) = subsecuencia(w, w')$.
- Si $a \neq b$, $subsecuencia(wa, w'b) = subsecuencia(w, w'b)$.

3. Una *celebridad* es una persona que no conoce a nadie (salvo a sí misma) y a quien todas conocen.

   Dadas n personas, se tiene una matriz C de $n \times n$, donde $C_{i,j} = 1$ sii la persona i conoce a la persona j, y $C_{i,j} = 0$ si no.

   Diseñe un algoritmo por inducción que encuentre una celebridad, si es que existe, en tiempo $O(n)$.

   Hint: Haga una primera pasada encontrando un único candidato a celebridad, y una segunda pasada comprobando si es realmente una celebridad. Parta demostrando por inducción sobre i que, después de haber analizado las personas 1 a i, puede descartar a todas menos a una como candidato a celebridad.

   Solución: Hacemos una primera pasada para obtener un único candidato a celebridad. Demostremos que eso es posible. Para $i = 1$, ese es el único candidato a celebridad. Para $i + 1$, supongamos por HI que k es el único candidato a celebridad en $[1..i]$. Entonces nos preguntamos si $i + 1$ conoce a k. Si lo conoce, entonces $i + 1$ no puede ser celebridad y k aún puede serlo. Si no lo conoce, entones k no puede ser celebridad e $i + 1$ es nuestro nuevo (y único) candidato a celebridad. Note que $i + 1$ podría no ser una celebridad, pero entre 1 e $i + 1$, sabemos que nadie más puede serlo.

   Una vez que encontramos un único candidato k en $[1..n]$, verificamos que todos lo conozcan y él no conozca a nadie.

   Se espera un seudocódigo para el algoritmo resultante, que es evidentemente $O(n)$. Pero si describen el algoritmo claramente de otra forma, también es aceptable.