CC3101 Matemáticas Discretas para la Computación
Profesor: Pablo Barceló B.
Auxiliares: Ilana Mergudich T.
Arniel Labrada D.
Fecha: Miércoles 2 de mayo de 2018

Auxiliar 6: Inducción Estructural y Algoritmos Recursivos

P1. *[C2 2013-1]* Recuerde la definición recursiva del inverso w^{-1} del string $w \in \Sigma^*$ vista en clases: Si $w = \epsilon$ entonces $w^{-1} = w$, y si $w = x \cdot a$, donde $x \in \Sigma^*$ y $a \in \Sigma$, entonces $w^{-1} = a \cdot x^{-1}$. Demuestre que para cada $w \in \Sigma^*$ se tiene que $(w^{-1})^{-1} = w$. Utilice sólo la definición de inverso dada arriba y la de concatenación vista en clases. Cualquier otra propiedad que utilice debe demostrarla.

P2. *[C2 2009-2]* El conjunto B de los strings de paréntesis balanceados se define recursivamente como sigue: (1) El string vacío ϵ está en B; y (2) si $x, y \in B$ entonces (x) y xy pertenecen a B.
Definimos la función N en el conjunto de strings de paréntesis de la siguiente forma:

$$N(\epsilon) = 0;\ N(() = 1;\ N()) = -1$$

$$N(uv) = N(u) + N(v)$$

a) Demuestre usando inducción estructural que si un string de paréntesis w es balanceado, entonces $N(w) = 0$ y $N(u) \geq 0$ para todo *prefijo* u de w, i.e. para todo u tal que $w = uu'$.

b) Demuestre usando inducción estructural que si un string de paréntesis w satisface $N(w) = 0$ y $N(u) \geq 0$ para todo *prefijo* u de w, entonces w es balanceado.

P3. *[C2 2015-1]* Asuma que tenemos n tareas $t_1, ..., t_n$ y que con cada tarea t_i $(1 \leq i \leq n)$ asociamos la siguiente información:

- El momento $c_i \geq 0$ en que la tarea comienza y otro $f_i > c_i$ en que finaliza.
- El beneficio $b_i > 0$ de que la tarea se ejecute.

Decimos que las tareas t_i y t_j son compatibles si sus intervalos de ejecución no se intersectan; es decir, si $f_i < c_j$ o $f_j < c_i$.
Asuma ahora que las tareas $t_1, ..., t_n$ se hallan ordenadas no decrecientemente con respecto a los términos de finalización de tareas; es decir, para todo $1 \leq i \leq j \leq n$ se cumple que $f_i \leq f_j$.
Defina $B : \{0, ..., n\} \to \mathbb{N}$ de tal forma que $B(i)$ corresponde al mayor beneficio que se puede obtener al ejecutar alguna de las i primeras tareas de forma compatible. Formalmente:

$$B(i) = \text{máx}\{\sum_{t_j \in A} b_j \mid A \subseteq \{t_1, ..., t_i\} \text{ y no hay dos tareas distintas en A que sean incompatibles}$$

a) Defina recursivamente a $B(i)$ en términos de $B(i-1)$ y $B(pred(i))$, donde $pred(i)$ es el mayor $j < i$ tal que t_j y t_i son compatibles.

b) De lo anterior, ¿qué podemos concluir con respecto al costo del problema de computar un conjunto de tareas mutuamente compatibles que maximicen el beneficio?

P4. *[C2 2016-1]* Cuando un corrector ortográfico encuentra un error en una palabra u utiliza una medida que le permite determinar cuáles son las palabras más "parecidas.ª u. Para esto utiliza una noción de distancia entre las palabras u y v que corresponde al menor número de *operaciones de edición* que son necesarias para convertir a u en v (estas operaciones se aplican simultáneamente en u). Las operaciones de edición permitidas son las siguientes: (a) insertar una letra, (b) borrar una letra y (c) cambiar a una letra por otra. Utilizamos la notación $d(u, v)$ para referirnos a la distancia entre u y v. Por ejemplo, si u=horda y v=ondas, entonces $d(u, v) = 3$ (borramos h, reemplazamos r por n, y agregamos s al final).

a) ¿Cuál es el valor de $d(u,v)$ cuando u o v corresponden a la palabra vacía?

b) Asuma que ni u ni v corresponden a la palabra vacía, es decir $u = a_1...a_m$ y $v = b_1...B_n$ donde $m, n > 0$ y los a_i's y b_j's son letras. Defina $u' = a_1...a_{m-1}$ y $v' = b_1...b_{n-1}$. Demuestre que

$$d(u,v) = \text{mín}\{1 + d(u',v), 1 + d(u,v'), diff(a_m,b_n) + d(u',v')\}$$

donde $diff(a_m, b_n) = 1$ si $a_m \neq b_n$ y $diff(a_m, b_n) = 0$ en caso contrario.
Ayuda: Represente la transformación de u a v de la siguiente forma: Si la letra a en la posición i no fue tocada, entonces ponemos el símbolo (a,a) en la posición i; si la letra a en la posición i fue reemplazada por b, entonces ponemos el símbolo (a,b) en la posición i; si la letra a en la posición i fue eliminada, entonces ponemos el símbolo $(a,)$ en la posición i; finalmente, si el símbolo a fue insertado en la posición j, entonces ponemos el símbolo $(,a)$ en la posición j. Por ejemplo, la transformación de u =horda a v =onda arriba descrita es representada por la palabra $(h,\_)(o,o)(r,n)(d,d)(a,a)(\_,s)$. Analice los diferentes casos que pueden ocurrir con respecto al símbolo que aparece en la última posición de la palabra que representa a la transformación de u a v.

c) Utilizando lo anterior, diseñe un algoritmo recursivo eficiente que calcule $d(u,v)$ y encuentre una secuencia mínima de operaciones necesarias para convertir a u en v.

P5. *[C2 2015-1]* Sea R_n el número de regiones que se generan en la superficie de una esfera al ser dividida por n grandes círculos (es decir por n planos que pasan por el centro de la esfera), asumiendo que no hay tres de estos grandes círculos que pasen por el mismo punto.

a) Explique por qué $R_{n+1} = R_n + 2n$, para $n \geq 2$.

b) Resuelva iterativamente la expresión anterior.