Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación
Universidad de Chile
CC3101-01 Matemáticas de Discoteca para la Computación

Profesor: Pablo Barceló
Auxiliares: Tomás Martínez, Ignacio Riego

Auxiliar 5

Jueves 4 de Mayo, 2017

P1. Sea Σ un alfabeto finito y Σ^* el conjutno de las palabras finitas formadas or símbolos en Σ. Dada una palabra $w = a_1...a_n$ *, se define su palabra *reversa* por $w^R = a_n...a_1$.

a) Dé una definición recursiva del operador reversa.

b) Muestre que para todo $w_1, w_2 \in \Sigma^*$ se tiene que $(w_1 \dot{w}_2)^R = w_2^R \dot{w}_1^R$

c) Dé una definición recursiva del operador potencia $w^i = w \dots w$, donde w está i veces, e $i \in N$

d) Muestre que $\forall i \in N, w \in \Sigma^*$ se cumple que $l(w^i) = i\dot{l}(w)$, donde $l(w)$ es el largo de w.

e) Muestre que $\forall i \in N, w \in \Sigma^*$ se cumple que $(w^i)^R = (w^R)^i$

P2. Imaginemos que un mechón o mechona quiere imprimir una serie de guías para estudiar, pero tiene tan solo k hojas para imprimir en el CEC. Cada guía g_i tiene una cantidad de hojas h_i que requiere para imprimirse y un valor asociado v_i que representa la cantidad de ayuda al estudio que va a aportar. Considerando que cada guía debe imprimirse completa o no imprimirse, defina un algoritmo recursivo que permita al o la estudiante maximizar el valor total de las guías que imprime.

P3. Definimos el arbol Stern-Brocot de la siguiente forma:

Casos base:

Definimos la raiz del arbol r como un nodo que tiene un valor asociado $\frac{1}{1}$. Esta no tiene padres y tiene un hijo derecho y un hijo izquierdo.

Caso inductivo:

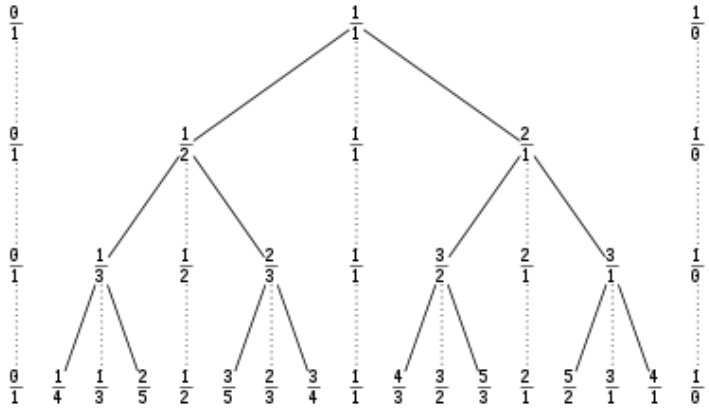
Dado un nodo a_p con valor asociado $\frac{m_p}{n_p}$ este tiene un hijo izquierdo y uno derecho.

Si un nodo a_{hd} es un hijo derecho de a_p y a_D es el ancestro más cercano de a_{hd} que está a su derecha. El valor asociado a a_{hd} es $\frac{m_p+m_D}{n_p+n_D}$.

Si no hay ningún ancestro a la derecha se toma $m_D = 1, n_D = 0$

De forma similar un nodo a_{hi} es un hijo izquierdo de a_p y a_I es el ancestro más cercano de a_h que está a su izquierda. El valor asociado a a_{hi} es $\frac{m_p+m_I}{n_p+n_I}$.

Si no hay ningún ancestro a la derecha se toma $m_I = 1, n_I = 0$



Diseñe un algortimo que dada una fracción $\frac{a}{b}$ irreductible encuentre el nodo dentro del arbol que la tiene como valor asociado. Puede asumir que toda fracción está en el arbol, sin demostrarlo.

P4. Asuma que tenemos n tareas $t_1, ..., t_n$, y a 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 $B : 0, ..., n \to N$ de tal forma que $B(i)$ corresponde al mayor beneficio que se puede obtener al ejecutar algunas de las i primeras tareas de forma compatible. Formalmente:

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

Defina recursivamente a B(i) en términos de $B(i1)$ y $B(pred(i))$, donde $pred(i)$ es el mayor $j <$ i tal que t_j y t_i son compatibles.