Jhonatan Rivera Saumeth

Taller 1 - Algoritmos

1) Ejercicios Cormen:

a) 3.1-2)

Para cualquier reales a y b donde $b > 0$

$(n+a)^b = \Theta(n^b)$

Si $c = 2^b$ y $n_0 \geq 2a \Rightarrow n \geq n_0$

es decir se tiene que $(n+a)^b \leq (2n)^b = cn^b$

entonces $(n+a)^b = O(n^b)$.

Si $n_0 \geq \frac{-a}{1-1/2^{1/b}}$ y $c = \frac{1}{2}$. Then $n > n_0 \geq \frac{-a}{1-1/2^{1/b}} \Longleftrightarrow n - \frac{n}{2^{1/b}} \geq -a$

$\Rightarrow n+a \geq (1/2)^{a/b} n \Rightarrow (n+a)^b \geq cn^b$. por definición $(n+a)^b = \Omega($

Se tiene que $f(n) = \Theta(g(n)) \Longleftrightarrow f(n) = O(g(n))$

y $f(n) = \Omega(g(n))$ como consecuencia de ello $(n+a)^b = \Theta(n$

b) 3.1-7)

Pruebe que $o(g(n)) \cap \omega(g(n))$ es el conjunto vacío.

Suponiendo que exista una $f(n)$ tal que $f(n) \in (o(g(n)) \cap \omega(g(n)))$

se tiene que $0 = \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$

Llegamos a una contradicción por lo tanto este conjunto no tiene elemento alguno.

c) 3.3) a) Ordenar por crecimiento asintótico:

$\lg(\lg^* n)$, $2^{\lg^* n}$, $\sqrt{2}^{\lg n}$, n^2, $n!$, $(\lg n)!$, $(\frac{3}{2})^n$, n^3, $\lg^2 n$
$\lg(n!)$, 2^{2^n}, $n^{1/\lg n}$, $\ln \ln n$, $\lg^* n$, $n \cdot 2^n$, $n^{\lg \lg n}$, $\ln n$, 1
$2^{\lg n}$, $(\lg n)^{\lg n}$, e^n, $4^{\lg n}$, $(n+1)!$, $\sqrt{\lg n}$, $\lg^*(\lg n)$, $2^{\sqrt{2 \lg n}}$
n, 2^n, $n \lg n$, $2^{2^{n+1}}$:

a) De mayor a menor:

$2^{2^{n+1}}$
2^{2^n}
$(n+1)!$
$n!$
$n2^n$
e^n
2^n
$(\frac{3}{2})^n$
$(\lg(n))!$
$n^{\lg(\lg n)}$
$\lg(n)^{\lg(n)}$
n^3
n^2
$4^{\lg(n)}$
$n\lg(n)$
$\lg(n!)$
$2^{\lg n}$
n
$\sqrt{2}^{\lg(n)}$
$2^{\sqrt{2\lg n}}$
$\lg^2 n$
$\ln(n)$
$\sqrt{\lg(n)}$
$\ln(\ln n)$
$2^{\lg^*(n)}$
$\lg^*(n)$
$\lg^*(\lg(n))$
$\lg(\lg^*(n))$
$n^{1/\lg(n)}$

b) de un ejemplo de una sola funcion no-negativa $f(n)$ tal que para todas las funciones $g_i(n)$ in la parte a), $f(n)$ ni es $O(g_i(n))$ ni $\Omega(g_i(n))$

Si se define la funcion:

$$f(n) = \begin{cases} g_1(n)! & n \bmod 2 = 0 \\ 1/n & n \bmod 2 = 1 \end{cases}$$

f es asintoticamente positiva:

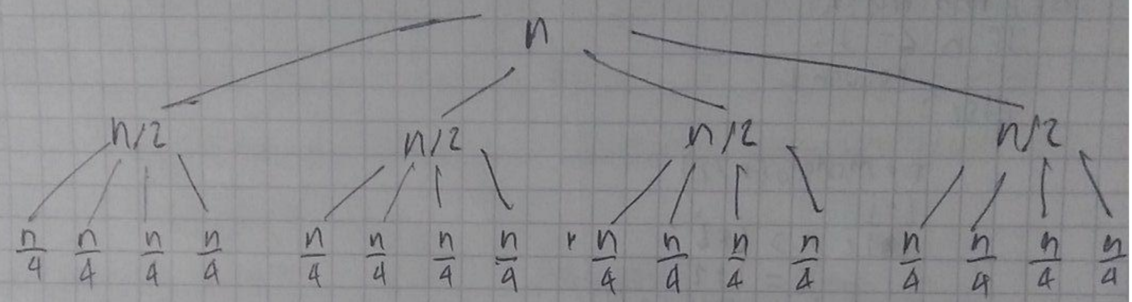
Para los pares: $\lim\limits_{n\to\infty} \frac{f(2n)}{g_i(2n)} \geq \lim\limits_{n\to\infty} \frac{f(2n)}{g_1(2n)}$
$= \lim\limits_{n\to\infty} (g_1(2n-$
$= \infty$

Para los impares: $\lim\limits_{n\to\infty} \frac{f(2n+1)}{g_i(2n+1)} \leq \lim\limits_{n\to\infty} \frac{f(2n}{1}$
$= \lim\limits_{n\to\infty} \frac{1}{2n+1}$
$= 0$

Se tiene que $f(n)$ no es $O(g_i(n))$ ni $f(n)$ es $\Omega(g_i(n))$ para todo i

d) 4.4-7) Dibuje el árbol de recursión para $T(n) = 4T(\lfloor n/2 \rfloor) + cn$ donde c es una constante y provee una cota asintótica en su solución, verifíquelo por el método de sustitución

Si $n = 4$



Una sustitución evidente es $\Theta(n^2) \Rightarrow T(n) \leq c'n^2$

$$t(n) = 4T(\lfloor n/2 \rfloor) + cn$$
$$\leq c'n^2 + cn$$

Lo que es menor que $c'n^2$ cuando $c' + \frac{c}{n} \leq 1$ Lo cual es cierto para n lo suficientemente grande

e) Usar método maestro para dar cotas ajustadas a las siguientes recurrencias:

* $T(n) = 8T(n/2) + n$ $\quad a = 8,\ b = 2,\ F(n) = n \quad c = 1$

$1 < \log_2 8$

$\Rightarrow T(n) = \Theta(n^3)$

* $T(n) = 8T(n/2) + n^3$ $\quad a = 8,\ b = 2,\ F(n) = n^3 \quad c = 3$

$3 = \log_2 8$

$\Rightarrow F(n) = \Theta(n^3 \log n)$

* $T(n) = 8T(n/2) + n^5$ $\qquad a = 8, b = 2, c = 5 \quad f(n) = n^5$

$5 > \log_2 8$

$\Rightarrow T(n) = \Theta(n^5)$

2) a) plantee una ecuacion de recurrencia para $T(n)$ el tiempo que toma la funcion misterio

```
def misterio(n):
    if n <= 1:
        return 1
    else:
        r = misterio(n/2)
        i = 1
        while n > i*i:
            i = i+1
        r = r + misterio(n/2)
        return r
```

$T(n) = 2T(n/2) + n^{1/2} + O(1)$

B) Dibuje el Arbol:

$cn^{1/2}$

$\frac{cn^{1/2}}{2} \qquad \frac{cn^{1/2}}{2}$

$\frac{cn^{1/2}}{4} \qquad \frac{cn^{1/2}}{4} \qquad \frac{cn^{1/2}}{4} \qquad \frac{cn^{1/2}}{4}$

. . . . . .

i) Altura: $\lg n$

ii) Nodos por nivel: 2^n

iii) suma de los nodos de cada nivel: $cn^{1/2}$

iv) Suma Total: $\lg n \cdot cn^{1/2}$

C) comportamiento asintótico:

$T(n) = 2T(n/2) + n^{1/2}$, $\qquad a = 2 \quad b = 2 \quad c = 1/2 \quad f(n) = \sqrt{n}$

$1/2 < \log_2 2$

$\Rightarrow T(n) = \Theta(n)$

$* \; T(n) = 8T(n/2) + n^5 \qquad a = 8,\ b = 2,\ c = 5 \quad f(n) = n^5$

$5 > \log_2 8$

$\Rightarrow T(n) = \Theta(n^5)$

22.3-1

Hacer un grafico 3 x 3 con etigoetas para columnas y filas Blanco, gris y negro. en cada celda (i, j) indicar si, en algun punto durante una busqueda en profundidad de un grafo dirigido puede haber un edge disde un nodo de color i a un vertice de color j. para cada posible edge indicar que tipo puede ser. hacer un segundo grafico tal que este sea la busqueda en profundidad para un grafo No-dirigido

a) Para el dirigido...

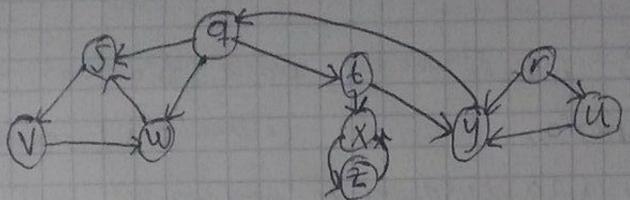
de \ hacia	Negro	gris	Blanco
Negro	T, B, F, C	B, C	B, C
Gris	C, T, F	T, F, B	B, C
Blanco	C, T, F	B C	T, B, F, C

Tree = T
Back = B
Forward = F
Cross = C

b) Para el no dirigido...

de \ hacia	Negro	Gris	Blanco
Negro	T, B, C, F	T, B, C, F	T, B, C, F
Gris		T, F, B	T, B, C, F
Blanco			T, B, C, F

22.3-2) Muestre como una busqueda en profundidad funciona en el grafo de la figura 22.6 Asuma que el Loop For de las lineas 5-7 considera los vertices en orden alfabetico.



nodo	Descubierto	finalizado
q	1	16
r	17	20
s	2	7
t	8	15
u	18	19
v	3	6
w	4	5
x	9	12
y	13	14
z	10	11

momentos en su descubierto y Finalizarse.

22.4-2 dado un algoritmo de tiempo lineal, que toma como entrada un grafo dirigido sin ciclos $G = (V, E)$ y dos vertices s y t y retorna el numero de caminos simples de s a t en G.

```
if u == v then
     return 1
else if  u.paths ≠ NIL then
     return u.paths
else
     for each w ∈ Adj[u] do
          u.paths = u.paths + SIMPLEPATHS(w, v)
     end for
     return u.paths
end if.
```