CC40A: Diseño y Análisis de Algoritmos
Auxiliar N^o 1

Prof. Gonzalo Navarro
Aux. Carlos Bedregal

Abril 07 de 2010

1. **Cotas**: ¿Verdadero o Falso?
   a) $n^2 + 10000n \in O(n^2)$
   b) $n^2 - n \in \Omega(n^2)$
   c) $n^{2,9999} \in \Omega(n^3)$
   d) $n^{10}/10000 \in o(n^{10})$
   e) $n^2/\lg n \in o(n^2)$
   f) $n^2 \lg n \in \omega(n^2)$
   g) $(\lg n)^{\lg n} \in o(n^{\lg \lg n})$
   h) $2^{\lg n} \in \Theta(n)$
   i) $\lg(n^2) + \lg n \in \Theta(\lg n^3)$
   j) $2^{2n} \in O(2^n)$ (Demostrar)

2. Demostrar las siguiente serie geométricas:
   - $\displaystyle\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{2^i} = 1.$
   - $\displaystyle\lim_{n\to\infty} \sum_{i=1}^{n} i\frac{1}{2^i} = 2.$

3. **Huffman:** Usando la codificación de Huffman, determinar el número de bits necesario para representar los elementos $\{a, b, c, d, e, f, g, h, i, j\}$ considerando las siguientes listas de probabilidades:
   a) $P_1 = \langle 33, 1, 1, 15, 5, 8, 13, 17, 2, 5\rangle$, $\mathcal{H}(P_1) = 2{,}73$
   b) $P_2 = \langle 4, 22, 10, 25, 7, 2, 1, 3, 15, 11\rangle$, $\mathcal{H}(P_2) = 2{,}86$

4. **Hu-Tucker**: Para las mismas probabilidades, determinar el número de bits si usamos la codificación de Hu-Tucker. Discuta:
   a) ¿Qué ventajas y desventajas se obtiene frente a Huffman?
   b) ¿Para que nos sirve la Entropía?
   c) ¿Cómo se podría aplicar a problemas de búsqueda?

5. Resolver las siguiente recurrencias [lineales homogéneas]:
   - $u_0 = 0$, $u_1 = 1$, $u_n = 5u_{n-1} - 6u_{n-2}$ $(n \geq 2)$.
   - $u_0 = 1$, $u_1 = 0$, $u_n = 6u_{n-1} - 8u_{n-2}$ $(n \geq 2)$.

- $u_0 = 0$, $u_1 = 1$, $u_2 = 3$, $u_n - u_{n-1} = 4[(u_{n-1} - u_{n-2}) - (u_{n-2} - u_{n-3})]$ $(n \geq 3)$.
- $u_0 = 1$, $u_1 = 2$, $u_2 = 3$, $u_n = 5u_{n-1} - 8u_{n-2} + 4u_{n-3}$ $(n \geq 3)$.

6. Resolver las siguiente recurrencias [teorema maestro]:

   - $T(n) = 2T(n/2) + n^3$.
   - $T(n) = 5T(n/2) + \Theta(n^2)$.
   - $T(n) = 27T(n/3) + \Theta(n^3 \lg n)$.
   - $T(n) = T(9n/10) + n$.
   - $T(n) = 7T(n/3) + n^2$.
   - $T(n) = 2^n T(n/4) + \sqrt{n}$.
   - $T(n) = T(\sqrt{n}) + 1$.

7. Si tenemos que escoger uno de los siguientes tres algoritmos:

   - Algoritmo X que resuelve el problema dividiéndolo en cinco subproblemas de la mitad del tamaño, resolviendo recursivamente cada uno de ellos para luego combinar las soluciones en tiempo lineal.
   - Algoritmo Y que resuelve un problema de tamaño n dividiéndolo en nueve subproblemas de tamaño $n/3$, resolviendo recursivamente cada uno y combinando las soluciones en tiempo $O(n^2)$.
   - Algoritmo Z que resuelve problemas de tamaño n recursivamente resolviendo dos subproblemas de tamaño $n - 1$ y luego combina las soluciones en tiempo constante.

   ¿Cuál es el tiempo de ejecución de cada uno? ¿Cuál escogerías?

8. ¿Cuántas líneas imprime la siguiente función?

   ```
   función f(n)
       if (n > 1) then
           imprimirLinea("una línea mas")
       f(n/2)
       f(n/2)
   ```

9. Imagina que tenemos un arreglo de enteros de tamaño infinito con las n primeras posiciones ordenadas (el resto de posiciones está llena con un símbolo #). Describe un algoritmo que encuentre la posición de un entero x (si existe) en tiempo $O(\log n)$.