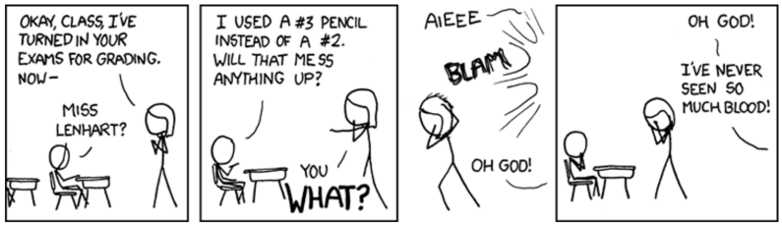
Segundo Certamen
Algoritmos y Complejidad

23 de diciembre de 2017

Scantron



1. Analice el tiempo de ejecución del algoritmo 1, llamado StoogeSort en honor a los inmortales Curly, Larry y Moe.

Algoritmo 1: Stooge Sort

procedure StoogeSort($a, i = 0, j = a.\text{length}() - 1$)
  if $a[i] > a[j]$ **then**
    tmp $\leftarrow a[i]; a[i] \leftarrow a[j]; a[j] \leftarrow$ tmp
  end
  if $j - i + 1 > 2$ **then**
    $k \leftarrow \lfloor (j - i + 1)/3 \rfloor$
    StoogeSort($a, i, j - k$)
    StoogeSort($a, i + k, j$)
    StoogeSort($a, i, j - k$)
  end

(25 puntos)

2. Para cierto problema de tamaño n cuenta con tres algoritmos alternativos:

Algoritmo A: Ordena los datos dados, divide en tres problemas de un tercio de tamaño y combina en tiempo constante.

Algoritmo B: Resuelve recursivamente dos problemas de tamaño $n-1$ y combina las soluciones en tiempo constante.

Algoritmo C: Divide el problema de tamaño n en nueve de tamaño $n/4$ y combina las soluciones en tiempo $O(n^2)$.

¿Cuál elije si n es grande, y porqué?

(35 puntos)

3. Una estructura de datos da un costo máximo total de $O(n \log n)$ al efectuar n operaciones. ¿Cuál es el costo amortizado por operación? ¿Cuál es el costo máximo por operación? ¿Cuál es el costo mínimo? ¿En qué situaciones debiera aplicarse el criterio tradicional de rendimiento en el peor caso y no costo amortizado?

(25 puntos)

4. El problema MASS (*Maximum Alternating Subset Sum*) busca una secuencia de índices $0 \le i_1 < i_2 < \cdots < i_k \le n-1$ tales que $a[i_1] - a[i_2] + a[i_3] - \cdots \pm a[i_k]$ sea máximo para un arreglo $a[n]$ dado. Por ejemplo, para $[4, 9, 2, 4, 1, 3, 7]$ es $9 - 2 + 4 - 1 + 7 = 17$. Dé un algoritmo basado en programación dinámica para hallar la suma de MASS (no se piden los índices), indicando los arreglos usados (conviene separar sumas alternantes de largo par e impar) y la recurrencia usada por su programa. Indique los valores iniciales, el valor buscado y el orden en que se efectúan los cálculos. ¿Cuál es el tiempo de ejecución aproximado en términos de n?

(40 puntos)

Torpedo

Teorema maestro

Al dividir un problema de tamaño n en a problemas de tamaño n/b que se resuelven recursivamente, si el trabajo de dividir y combinar soluciones es $f(n) > 0$, el tiempo para resolver un problema de tamaño n es:

$$t(n) = a t(n/b) + f(n) \quad t(1) = t_1$$

cuya solución es:

$$t(n) = \begin{cases} \Theta(n^{\log_b a}) & f(n) = O(n^c) \text{ para } c < \log_b a \\ \Theta(n^{\log_b a}) & f(n) = \Theta(n^{\log_b a} \log^\alpha n) \text{ con } \alpha < -1 \\ \Theta(n^{\log_b a} \log\log n) & f(n) = \Theta(n^{\log_b a} \log^\alpha n) \text{ con } \alpha = -1 \\ \Theta(n^{\log_b a} \log^{\alpha+1} n) & f(n) = \Theta(n^{\log_b a} \log^\alpha n) \text{ con } \alpha > -1 \\ \Theta(f(n)) & f(n) = \Omega(n^c) \text{ con } c > \log_b a \text{ y } a f(n/b) < k f(n) \text{ para } n \text{ grande con } k < 1 \end{cases}$$

Algunas series notables

En lo que sigue, $m, n \in \mathbb{N}$, $k \in \mathbb{N}_0$ y $\alpha, \beta \in \mathbb{C}$.

$$\frac{1}{1 - az} = \sum_{n \geq 0} a^n z^n \qquad \frac{1 - z^{m+1}}{1 - z} = \sum_{0 \leq n \leq m} z^n$$

$$(1 + z)^\alpha = \sum_{n \geq 0} \binom{\alpha}{n} z^n \qquad \binom{\alpha}{n} = \frac{\alpha^{\underline{n}}}{n!} = \frac{\alpha(\alpha - 1) \cdots (\alpha - n + 1)}{n!} \qquad \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

$$\binom{1/2}{n} = \frac{(-1)^{n-1}}{n 2^{2n-1}} \binom{2n-2}{n-1} \quad (\text{si } n \geq 1) \qquad \binom{-1/2}{n} = \frac{(-1)^n}{2^{2n}} \binom{2n}{n} \qquad \binom{-n}{k} = (-1)^k \binom{n+k-1}{n-1}$$

$$\frac{1}{(1-z)^{n+1}} = \sum_{k \geq 0} \binom{n+k}{n} z^k = \frac{1}{n!} \sum_{k \geq 0} (k+1)(k+2) \cdots (k+n) z^k = \frac{1}{n!} \sum_{k \geq 0} (k+1)^{\bar{n}} z^k$$

$$\sum_{n \geq 0} \binom{n}{k} z^n = \frac{z^k}{(1-z)^{k+1}}$$