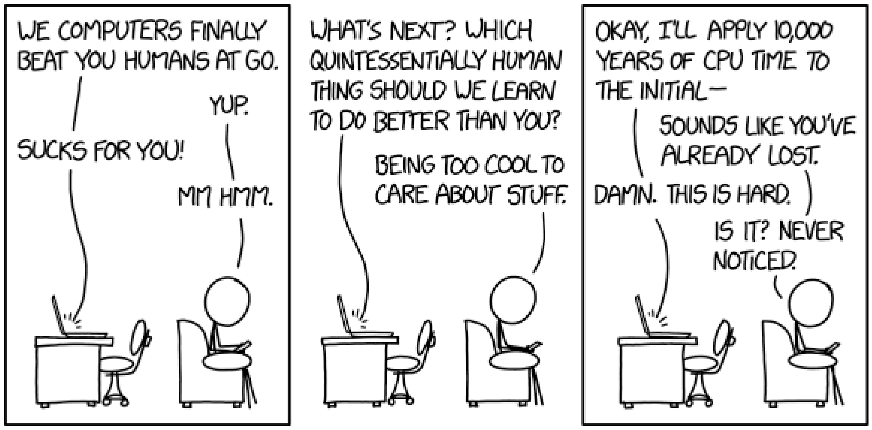
Rezago Segundo Certamen
Algoritmos y Complejidad

25 de agosto de 2017



1. Se da una colección de tareas con sus duraciones. Demuestre que si se ejecutan las tareas en orden de duración creciente, se obtiene el mínimo de la suma de los instantes de término de las mismas.

   (25 puntos)

2. Para resolver cierto problema de gran tamaño n, se plantean varios algoritmos basados en dividir y conquistar:

   Algoritmo A: Divide en tres problemas de la mitad del tamaño, dividir y combinar soluciones tiene costo $\Theta(\sqrt{n})$

   Algoritmo B: Divide en siete problemas de igual tamaño, costo extra es lineal en n.

   Algoritmo C: Divide el problema en dos problemas de igual tamaño, dividir y combinar cuesta $O(n \log n)$

   Algoritmo D: Descompone de tamaño n en dos de tamaño $n-1$, el costo de dividir y combinar es constante.

   Para problemas grandes, ¿cuál elige y porqué? ¿Requiere análisis más detallado?

   (35 puntos)

3. Un *heap* es un árbol binario casi completo en el cual cada elemento es menor que cada uno de sus hijos. El algoritmo 1 construye un *heap* del rango $a[i..j]$ de un arreglo a de n elementos. Se invoca SlowHeap$(0, n-1)$. Plantee la

Algoritmo 1: SlowHeap

function SlowHeap(i, j)
  if $i = j$ **then**
    return Nodo sin hijos conteniendo $a[i]$
  Busque m en $i \le m \le j$ tal que $a[m]$ es el mínimo del rango
  Intercambie $a[m]$ con $a[j]$
  $p_l \leftarrow$ SlowHeap$(i, \lfloor (i+j-1)/2 \rfloor)$
  $p_r \leftarrow$ SlowHeap$(\lfloor (i+j-1)/2 \rfloor + 1, j-1)$
  return Nodo conteniendo $a[j]$ con hijos p_l y p_r

   recurrencia que describe el tiempo de ejecución de SlowHeap con un arreglo de n elementos, y obtenga su tiempo de ejecución en orden de magnitud.

   (30 puntos)

4. Dadas denominaciones de monedas $d_0 = 1 < d_1 < \cdots < d_k$, el problema de cambio es hallar el mínimo número de monedas para entregar la cantidad n. Plantee un algoritmo eficiente para obtener el mínimo número de monedas requerido. Note que el algoritmo voraz de entregar lo que se pueda de la máxima denominación no siempre funciona, por ejemplo con denominaciones $1, 5, 20, 25$ para 40 da $25, 5, 5, 5$ (4 monedas), el óptimo es $20, 20$ (2 monedas).

   (30 puntos)

Teorema maestro

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

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

cuya solución es:

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

Algunas series notables

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

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

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

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

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

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

Probabilidades

Teorema (Linearidad del valor esperado)**.** *Sean X_1, X_2 variables aleatorias, α y β constantes arbitrarias. Entonces:*

$$\mathbb{E}[\alpha X + \beta Y] = \alpha \mathbb{E}[X] + \beta \mathbb{E}[Y]$$

Decimos que dos variables X e Y son *independientes* si para todo par de valores x e y:

$$\Pr[(X = x) \wedge (Y = y)] = \Pr[X = x] \cdot \Pr[Y = y] \tag{1}$$

Una colección de variables es *independiente a pares* si para todo $i \neq j$ y todo par de valores x_i, x_j:

$$\Pr[(X_i = x_i) \wedge (X_j = x_j) = \Pr[X_i = x_i] \cdot \Pr[X_j = x_j] \tag{2}$$

Una colección de variables es *mutuamente independiente* si para todo subconjunto $S \subseteq N$ y toda colección de valores x_i:

$$\Pr[(X_{i_1} = x_{i_1}) \wedge (X_{i_2} = x_{i_2}) \wedge \cdots \wedge (X_{i_s} = x_{i_s})] = \Pr[X_{i_1} = x_{i_1}] \cdot \Pr[X_{i_2} = x_{i_2}] \cdots \Pr[X_{i_s} = x_{i_s}] \tag{3}$$