TEORIA DE ALGORITMOS

Segundo de Ingenieria Superior Informatica

Examen de Septiembre de 1997

1. Suponiendo la notacion habitual para la eficiencia de algoritmos, demostrar la veracidad o falsedad de la siguiente equivalencia

$$\lim_{n \to \infty} f(n)/g(n) \in \mathbf{R}^+ \Leftrightarrow f(n) \text{ es } \mathbf{O}(g(n) \text{ y } g(n) \text{ es } \mathbf{O}(f(n)$$

indicando si las dos implicaciones son ciertas, falsas, o solo alguna de ellas es cierta.

2. Demostrar que para el problema de la mochila continuo puede definirse una funcion de seleccion tal que produzca un algoritmo "greedy" que siempre encuentre una solucion optima. ¿Cual sera la eficiencia de ese algoritmo?.

3. Definir el caso peor y el caso promedio de un algoritmo. Como aplicacion, considerese un vector de tamaño n, A, constituido por elementos de la misma naturaleza. Sea T(m) el tiempo medio que requiere una llamada al conocido algoritmo "Quicksort" para un sub-vector A[i..j], donde m = j-i+1 es el numero de elementos que hay en el correspondiente sub-vector. Escribir y justificar una ecuacion para calcular el tiempo medio requerido para ejecutar "Quicksort" en el vector A. ¿Que tiempo consumira la operacion de pivoteo que se supone realiza el algoritmo?.

4. Definir lo que se entiende por Juego. Identificar cada una de las caracteristicas de esa definicion sobre el Juego de los Palillos, o de Nim. Explicar en que consiste la Regla Minimax. Escribir y justificar todas las formas que conozca para calcular el valor de una configuracion en un juego, y si hay mas de una demostrar que proporcionan el mismo valor.

5. Metodos de calculo de la eficiencia para algoritmos "backtracking". ¿En que consiste el problema del coloreo de un grafo?. Diseñar un algoritmo backtracking para ese problema. ¿Que se puede decir de su eficiencia?. ¿Seria mas aconsejable emplear un algoritmo "greedy" para resolver este problema?

1. Suponiendo la notacion habitual para la eficiencia de algoritmos, demostrar la veracidad o falsedad de la siguiente equivalencia

$$\lim_{n\to\infty} f(n)/g(n) \in \mathbf{R}^+ \Leftrightarrow f(n) \text{ es } \mathbf{O}(g(n)) \text{ y } g(n) \text{ es } \mathbf{O}(f(n))$$

indicando si las dos implicaciones son ciertas, falsas, o solo alguna de ellas es cierta.

$\boxed{\Rightarrow}$ $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = L \in \mathbb{R}^+$

$\left|\dfrac{f(n)}{g(n)} - L\right| \leq \varepsilon$

$\forall \varepsilon > 0, \exists m \in \mathbb{N}$ tq $\forall n \in \mathbb{N}$ con $n \geq m$

$$-\varepsilon \leq \frac{f(n)}{g(n)} - L \leq \varepsilon$$

$$L - \varepsilon \leq \frac{f(n)}{g(n)} \leq L + \varepsilon$$

$$g(n)(L-\varepsilon) \leq \underbrace{f(n) \leq (L+\varepsilon)\, g(n)}_{f(n) \in O(g(n))}$$

$\lim\limits_{n\to\infty} \dfrac{g(n)}{f(n)} = \dfrac{1}{L} = L' \in \mathbb{R}^+ \Rightarrow$ Y el razonamiento sería análogo, llegando a que $g(n) \in O(f(n))$

$\boxed{\Leftarrow}$ $f(n) \in O(g(n)) \Rightarrow \exists n_0 \in \mathbb{N} / \forall n \geq n_0\ \exists c \in \mathbb{R}^+$ con $f(n) \leq c \cdot g(n)$

$g(n) \in O(f(n)) \Rightarrow \exists n_0' \in \mathbb{N} / \forall n \geq n_0'\ \exists d \in \mathbb{R}^+$ con $g(n) \leq d \cdot f(n)$

$$c = \lim_{n\to\infty} \frac{f(n)}{\frac{1}{c}\cdot f(n)} \leq \lim_{n\to\infty} \frac{f(n)}{g(n)} \leq \lim_{n\to\infty} \frac{c\cdot g(n)}{g(n)} = c$$

2. Demostrar que para el problema de la mochila continuo puede definirse una funcion de seleccion tal que produzca un algoritmo "greedy" que siempre encuentre una solucion optima. ¿Cual sera la eficiencia de ese algoritmo?.

La función de selección en cuestión escoge siempre aquel objeto cuyo precio por unidad de peso sea mayor.

Lista candidatos: Objetos
Lista usados: Objetos ya introducidos
Criterio factibilidad: La fracción del objeto que tomamos sumada al peso acumulado no supera M (capac. mochila).
Criterio solución: La lista de objetos selec. pesa, como mucho, M. $\sum w_i x_i \leq M$
Función objetivo: máx $\sum p_i x_i$
Función selección: Introducir el objeto de mayor $A_i = \frac{p_i}{w_i}$

Suponiendo que tenemos ordenados los objetos por A_i y por tanto, nuestra solución será de la forma:

$$1 \quad 1 \quad \cdots \quad 1 \quad x_k \quad 0 \quad \cdots \quad 0 \qquad 0 \leq x_k \leq 1$$

Consideramos otra solución cualquiera $\{y_i\}$ y necesitamos comprobar que $\sum_{i=1}^{n} (x_i - y_i) p_i \geq 0$

$$\sum_{i=1}^{n} w_i (x_i - y_i) \frac{p_i}{w_i} = \sum_{i=1}^{k-1} w_i (x_i - y_i) \frac{p_i}{w_i} + w_k (x_k - y_k) \frac{p_k}{w_k}$$

$$+ \underbrace{\sum_{i=k+1}^{n} w_i (\underset{\overset{\shortparallel}{0}}{x_i} - y_i) \frac{p_i}{w_i}}_{<0} \geq \sum_{i=1}^{k-1} w_i (x_i - y_i) \frac{p_k}{w_k} + w_k (x_k - y_k) \frac{p_k}{w_k}$$

$$+ \sum_{i=k+1}^{n} w_i (x_i - y_i) \frac{p_k}{w_k} = \frac{p_k}{w_k} \sum_{i=1}^{n} w_i (x_i - y_i)$$

Y como $\sum w_i x_i = M$ y $\sum w_i y_i \leq M$, hemos acabado

En cuanto a la eficiencia, primero se calculan todos los A_i que sería $O(n)$, luego se ordena de menor a mayor con quicksort $O(n \log n)$ y por último vamos añadiendo o no, que es $O(n)$.