Metodología y tecnología de la programación

Ingeniería Técnica en Informática de Sistemas (UCM)

Hoja de ejercicios 8 Curso 2008/2009

Ejercicios de complejidad computacional

Ejercicio 1 Demostrar que el problema de la ordenación de un vector basado en comparaciones tiene una complejidad en $\Omega(n \lg n)$.

Ejercicio 2 Sea el problema de determinar si un grafo no dirigido con n vértices contiene un camino de longitud al menos 2.

1. Mediante el método del adversario, demostrar que cualquier algoritmo que resuelva este problema tiene un coste en $\Omega(n^2)$ en el peor caso si está restringido a realizar preguntas de la forma "¿Existe una arista entre los vértices i y j?". (Por ejemplo, si el grafo se representa mediante una matriz de adyacencia.)
2. Demostrar que, sin embargo, este problema se puede resolver en tiempo $O(n)$ si el algoritmo puede preguntar en cada vértice por la lista de sus vértices adyacentes.

Ejercicio 3 Utilizar el método del adversario para demostrar que cuando todos los elementos de un vector son distintos no es posible localizar la mediana con certeza sin mirar a cada elemento. Por otro lado, mostrar con un ejemplo que este no es el caso si los elementos no son distintos.

Ejercicio 4 Demostrar que las reducciones polinómicas son transitivas. Sean A, B y C tales que $\mathsf{A} \leq^T \mathsf{B}$ y $\mathsf{B} \leq^T \mathsf{C}$; demostrar que $\mathsf{A} \leq^T \mathsf{C}$. Repetir la demostración para $\leq^p$ con la restricción de que A, B y C sean problemas de decisión.

Ejercicio 5 Encontrar dos problemas de decisión sencillos, X e Y, tales que $X \leq^T Y$ pero $X \not\leq^p Y$.

Ejercicio 6

1. Sean X e Y dos problemas de decisión. Demostrar que si $X \in \mathcal{NP}$ y $Y \leq^p X$ entonces $Y \in \mathcal{NP}$.
2. Mostrar evidencia convincente de que es posible que $X \in \mathcal{NP}$ y $Y \leq^T X$, y sin embargo $Y \notin \mathcal{NP}$ incluso aunque Y sea un problema de decisión.

Ejercicio 7 Demostrar que el problema de encontrar una asignación que satisfaga una fórmula booleana es Turing equivalente en tiempo polinómico al problema de decidir si tal asignación existe.

Ejercicio 8 Demostrar que la versión de decisión del problema del viajante es Turing equivalente en tiempo polinómico a la versión de optimización.

Ejercicio 9 Sean los tres problemas siguientes:

1. PDC: Dado un grafo G y un natural n, ¿existe un clique de tamaño n en G?
2. POC: Dado un grafo G, encontrar el tamaño del mayor clique en G.
3. PC: Dado un grafo G, encontrar un clique de tamaño máximo en G.

Demostrar que los tres son Turing equivalentes en tiempo polinómico.

Ejercicio 10 Mostrar mediante un ejemplo que el algoritmo voraz que resuelve el problema de la mochila puede dar lugar a soluciones arbitrariamente malas si se aplica en la versión en que los objetos no se pueden partir. Modificar ligeramente dicho algoritmo para obtener un algoritmo 2-aproximado que resuelva el problema. (Se puede suponer que ningún objeto pesa más que la capacidad de mochila.)