Métodos Algorítmicos en Resolución de Problemas

Grado en Ingeniería Informática

Hoja de ejercicios 9 Curso 2018–2019

Ejercicios de complejidad computacional

Ejercicio 1 Consideremos 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 2 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 3 Demostrar que las reducciones Turing-polinómicas son transitivas. Sean A, B y C tales que $A \leq^T B$ y $B \leq^T C$; demostrar que $A \leq^T C$. Repetir la demostración para $\leq^p$ con la restricción de que A, B y C sean problemas de decisión.

Ejercicio 4 Encontrar dos problemas de decisión sencillos, X e Y, tales que $X \leq^T Y$ pero $X \not\leq^p Y$.

Ejercicio 5 Decimos que dos problemas A y B son Turing-equivalentes si $A \leq^T B$ y $B \leq^T A$. 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 6 Demostrar que la versión de decisión del problema del viajante es Turing equivalente en tiempo polinómico a la versión de optimización.

Ejercicio 7 Sean los tres problemas siguientes:

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

Demostrar que los tres son Turing equivalentes en tiempo polinómico.

Ejercicio 8

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

Ejercicio 9 Un conjunto independiente en un grafo $G = (V, A)$ es un subconjunto de vértices $V' \subseteq V$ tal que cada arista en A incide como mucho en un vértice de V'. El problema del conjunto independiente (PCI) consiste en encontrar un conjunto independiente de tamaño máximo en G. Formular el problema de decisión correspondiente y demostrar que es NP-completo. *Sugerencia*: utilizar el problema de decisión del clique.

Ejercicio 10 El problema de la **Suma** consiste en decidir, dados un vector de enteros $A[1..n]$ y un valor S, si podemos encontrar un subconjunto I del conjunto de índices del vector tal que la suma de los correspondientes elementos del mismo sea S, es decir, $\sum_{i \in I} A[i] = S$.

El problema de la **Partición** de un vector de enteros $A[1..n]$ consiste en decidir si podemos separar los elementos del vector en dos partes disjuntas que sumen lo mismo. O sea, nos preguntamos si existen I_1 e I_2 con $I_1 \cup I_2 = \{1, \ldots, n\}$ e $I_1 \cap I_2 = \emptyset$ tales que $\sum_{i \in I_1} A[i] = \sum_{i \in I_2} A[i]$.

Suponiendo demostrado que **Suma** es NP-completo, demostrar que **Partición** también lo es.

Ejercicio 11 Llamamos **Cabaña** al problema de decisión de la cabaña, que pide discernir si empalmando cañas de longitudes $l_1, \ldots, l_n$ es posible montar cuatro pilares, el menor de los cuales tenga una altura mayor o igual que un L dado.

Sabiendo que **Partición** es NP-completo, demostrar que **Cabaña** también lo es.

Ejercicio 12 El problema de decisión **Mochila** consiste en decidir, dados dos naturales M y B, un vector de pesos $p_1, \ldots, p_m$ y otro vector de beneficios $b_1, \ldots, b_m$, si existe un subconjunto de índices $S \subseteq \{1, \ldots, m\}$ tales que $\sum_{i \in S} p_i \leq M$ y a la vez $\sum_{i \in S} b_i \geq B$.

Sabiendo que **Partición** es NP-completo, demostrar que **Mochila** también lo es.

Ejercicio 13 Para $k > 1$ fijo, el problema de la **k-Repartición** consiste en decidir si podemos separar los elementos de un vector $A[1..n]$ de naturales en k subconjuntos disjuntos que sumen lo mismo.

Sabiendo que **Partición** es NP-completo, demostrar que todo problema **k-repartición**, fijado un $k > 1$, también lo es.

Ejercicio 14 Dados los naturales $b_1, \ldots, b_m, E$ y k, el problema **Envasar** consiste en decidir si es posible dividir los b_i en un número menor o igual que k de subconjuntos disjuntos tales que la suma de los elementos de cada subconjunto sea a lo sumo E.

1. Sabiendo que **Partición** es NP-completo, demostrar que **Envasar** también lo es.
2. Es más, aún fijado un $k > 1$ cualquiera a priori, el correspondiente problema k-**Envasar**, sigue siendo NP-completo. Demostrarlo.

Ejercicio 15 Tenemos un conjunto de números naturales cualesquiera $\{v_1, \ldots, v_n\}$. Nuestra tarea, dadas sendas cantidades S y T con $S < T$, será escoger un subconjunto de valores que en total sumen una cantidad V que cumpla $S \leq V < T$). Llamamos **Intervalo** al problema de decidir si esto es posible.

1. Demostrar que **Intervalo** es un problema NP-completo.
2. Complicamos ahora un poco el problema. Para cada $M > 0$ definimos un problema M-**Intervalo** que, dado un conjunto de naturales $\{v_1, \ldots, v_n\}$ y las cantidades S y T que cumplen $T - S \geq M$, decide si existe un subconjunto de $\{v_1, \ldots, v_n\}$ que sumen una cantidad cualquiera V tal que $S \leq V < T$.

   Aunque la tarea es aparentemente más fácil, demostrar que todo problema M-**Intervalo** sigue siendo NP-completo.

Sugerencia: Usar en ambos casos como problema de referencia NP-completo el problema **Suma**.

Ejercicio 16 El problema del camino hamiltoniano consiste en ver si existe un camino en un grafo dirigido dado G, que pasa una sola vez por cada vértice del mismo, y el del ciclo hamiltoniano en saber si partiendo de un vértice cualquiera podemos recorrer todos los vértices y volver al origen, sin pasar dos veces por ninguno de ellos. Considerando ahora probado que el problema del camino hamiltoniano es NP-completo, demostrar que se puede reducir al problema del circuito hamiltoniano.

Ejercicio 17 Nos dan una secuencia $x_1, \ldots, x_n$ de n números naturales cuyo orden no podremos alterar. Interesa decidir si es posible intercalar entre cada dos de ellos los signos $+$ o $-$, de manera que el valor de la expresión resultante, evaluada de izquierda a derecha, y sin ningún tipo de precedencia entre sus operadores, sea finalmente 0. Por ejemplo, dada la secuencia $8, 2, 4, 3, 1$ podríamos en este caso obtener 0 así: $8 - 2 - 4 - 3 + 1 = 6 - 4 - 3 + 1 = 2 - 3 + 1 = -1 + 1 = 0$.

Mostrar por medio de una reducción sencilla de alguno de los problemas NP-completos conocidos, que a pesar de su aparente simplicidad, el problema planteado también es NP-completo.

Ejercicio 18 El problema SAT-k-CNF es el problema de satisfactibilidad de una fórmula lógica del cálculo de proposiciones expresada en forma normal conjuntiva, tal que cada cláusula tiene a lo sumo k literales. Suponiendo demostrado que el problema SAT-4-CNF es NP-completo, demostrar que el problema SAT-3-CNF también lo es.

Ejercicio 19 Decimos que dos problemas de decisión A y B son p-equivalentes si $A \leq^p B$ y $B \leq^p A$. Demuestra que los tres problemas siguientes sobre grafos no dirigidos $\langle V, A\rangle$ son p-equivalentes:

PDCI (Problema de Decisión del Conjunto Independiente): determinar si un grafo tiene un conjunto independiente de tamaño al menos k. Un conjunto independiente es un subconjunto de vértices $V' \subseteq V$ tal que cada arista de A incide como mucho en un vértice de V'.

PDC (Problema de Decisión del Cliqué): determinar si un grafo tiene un cliqué (subgrafo completo) que contenga al menos k vértices.

PDCV (Problema de Decisión de la Cobertura de Vértices): determinar si un grafo tiene una cobertura de tamaño como mucho k. Una cobertura es un subconjunto de vértices $V' \subseteq V$ tal que si $\langle u, v\rangle \in A$, entonces $u \in V'$ o $v \in V'$ (o ambos).

Ejercicio 20 Dada una familia de conjuntos finitos $F = \{S_1, \ldots, S_n\}$, no necesariamente disjuntos, y dado un k, $1 \leq k \leq n$, el problema del *SET-COVER* consiste en decidir si existe una subfamilia $F' = \{S_{i_1}, \ldots, S_{i_{k'}}\}$, $F' \subseteq F$, de tamaño $k' \leq k$, que cubra F. Se entiende por ello que la unión de todos los conjuntos de F' ha de coincidir con la unión de todos los conjuntos de F. Sabiendo que el problema de la cobertura de vértices es NP-completo, demostrar que *SET-COVER* también lo es.

Ejercicio 21 Dado un conjunto finito S de objetos, una familia de conjuntos $F = \{S_1, \ldots, S_n\}$, todos ellos subconjuntos de S, y un número natural k, el problema de decisión del *hitting set* consiste en saber si existe un subconjunto $H \subseteq S$, de cardinal a lo sumo k, que tenga intersección no vacía con todos los conjuntos S_i de la familia F.

Sabemos que los problemas de decisión del *Cliqué*, de la *Cobertura de vértices* y del *Conjunto independiente* son NP-completos. Demostrar a partir de alguno de estos problemas que el problema del *hitting set* también lo es.