Ejercicios Tema 1

Algoritmia para problemas difíciles

Departamento de Informática e Ingeniería de Sistemas
Escuela de Ingeniería y Arquitectura – Universidad de Zaragoza

6 de octubre de 2017

Ejercicios sobre NP-difíciles

Ejercicio 1 Sea *GUERRA-HIPER-COMUNIDADES* el siguiente problema: Dada una gran colección de n páginas web y un número k se pretende saber si existen dos comunidades rivales de fuerza k, es decir, dos conjuntos disjuntos A_1, A_2 de k páginas web cada uno, de forma para dos páginas cualquiera x, y del mismo A_i la página x tiene un hipervínculo a la y (y viceversa). Demostrar que *GUERRA-HIPER-COMUNIDADES* es un problema intratable.

Ejercicio 2 En el problema *Medio-SAT* se nos da C una fórmula booleana CNF con n variables y m cláusulas, donde m es par. Queremos determinar si existe una asignación de verdad para las variables de C de tal manera que exactamente la mitad de las cláusulas evalúan a falso y exactamente la mitad de las cláusulas evalúan a cierto. Demostrar que *Medio-SAT* es intratable con una reducción desde *SAT*.
Notación: Un *literal* es una variable booleana afirmada o negada (p. ej. x, $\neg y$, etc). Una *cláusula* es una disyunción de literales (p. ej. $(x \vee \neg y)$). Una *fórmula en CNF* es una conjunción de cláusulas (p. ej. $(x \vee \neg y \vee z) \wedge (\neg z \vee y) \wedge (\neg x)$).
Pistas: Una fórmula puede tener cláusulas repetidas. Se pueden escribir cláusulas que siempre sean ciertas.

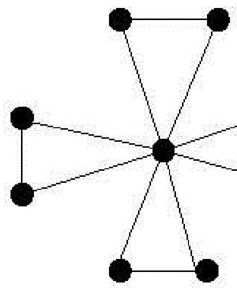
Ejercicio 3 En el problema *Medio-VC* se nos da G un grafo de n vértices y m aristas y un natural k. Queremos determinar si existe un conjunto de k vértices que cubra la mitad de las aristas del grafo. Demostrar que *Medio-VC* es intratable con una reducción desde *Cobertura de Vértices*.

Ejercicio 4 Supongamos que se nos da un programa que puede resolver el problema decisional *TSP* (transparencia 48) en, por ejemplo, tiempo lineal. Se pide un algoritmo eficiente para encontrar el recorrido óptimo de TSP con un número polinómico de llamadas a dicho programa.

Ejercicio 5 Diseñar e implementar un algoritmo de backtracking para probar si un circuito CNF es satisfacible. ¿Qué criterios se pueden utilizar para podar esta búsqueda?

Ejercicio 6 Un ejemplo del problema de *Cobertura por Conjuntos* se compone de un conjunto X de n elementos, una familia F de subconjuntos de X y un entero k. La pregunta es, ¿existen k subconjuntos de los de F cuya unión es X?
Por ejemplo, si $X = \{1, 2, 3, 4\}$ y $F = \{\{1, 2\}, \{2, 3\}, \{4\}, \{2, 4\}\}$, no existe una solución para $k = 2$, pero sí existe para $k = 3$ (por ejemplo, $\{1, 2\}, \{2, 3\}, \{4\}$). Demostrar que *Cobertura por Conjuntos* es intratable con una reducción desde *Cobertura de Vértices*.

Ejercicio 7 El problema *Árbol de recubrimiento de bajo grado* es el siguiente: dado un grafo G y un entero k, ¿contiene G un árbol de recubrimiento tal que todos los vértices del árbol tienen grado como mucho k (obviamente, sólo las aristas en el árbol cuentan para el grado)?
(Para la definición de árbol de recubrimiento o *spanning tree* se pueden ver el libro de Skiena o los apuntes de AB (tema 2, transparencia 40)).
Por ejemplo, en el siguiente grafo, no hay ningún de recubrimiento de tal manera que todos los vértices tienen un grado menor que tres.



1. Demostrar que problema *Árbol de recubrimiento de bajo grado* es intratable con una reducción desde *camino Hamiltoniano* (*camino Hamiltoniano* es un problema similar a *ciclo Hamiltoniano* pero en el que se busca un camino sin repeticiones y por todos los vértices que no tiene que ser cerrado).
2. Ahora considerar el problema *Árbol de recubrimiento de alto grado*, que es como sigue. Dado un grafo G y un entero k, ¿contiene G un árbol de recubrimiento tal que el mayor grado de un vértice es por lo menos k? En el ejemplo anterior, existe un árbol de recubrimiento con grado más alto 8. Dar un algoritmo eficiente para resolver el problema *Árbol de recubrimiento de alto grado*, y un análisis de su complejidad en tiempo.

Ejercicio 8 Demostrar que el siguiente problema es intratable:
Problema: Clique, independiente
Entrada: Un grafo G y un entero k.
Salida: ¿Contiene G un clique de k vértices y un conjunto independiente de k vértices?

Ejercicio 9 Demostrar que el siguiente problema es intratable:
Problema: Subgrafo denso
Entrada: Un grafo G y enteros k, y.
Salida: ¿Contiene G un subgrafo con exactamente k vértices y al menos y aristas?

Ejercicio 10 Explicar detalladamente un algoritmo en tiempo polinómico para resolver 2-SAT (ver definición en transparencias).

Ejercicio 11 Demostrar que los siguientes problemas están en NP:

- ¿El grafo G tiene un camino simple (es decir, sin repetir vértices) de longitud k?
- ¿Es n un entero compuesto (es decir, no es primo)?
- ¿Tiene un grafo G una cobertura de vértices de tamaño k?

En caso de entregar alguno de estos ejercicios, la fecha límite es el lunes 23 de octubre.
Antes de realizar cualquiera de estos ejercicios el alumno debe enviar un correo a `elvira@unizar.es` indicando qué ejercicio desea realizar.
Cualquier fuente utilizada en la resolución de estos ejercicios debe ser indicada claramente en la solución.