Análisis de Algoritmos
Segundo Problemario

Prof. Miguel A. Pizaña

9 de Julio de 2007

I Tareas

1. ¿Qué es el gato de Schrödinger? ¿El experimento de la doble rejilla? ¿Los experimentos de elección tardía (delayed choice experiments)? ¿El colapso de la función de onda? ¿La interpretación de Conpenhague? ¿La computación cuántica? ¿La teleportación cuántica? ¿Qué tiene que ver todo esto con las Máquinas de Turing No Deterministas? ¿Con los algoritmos no deterministas (i.e. con "choice")?

2. ¿Quiénes son y cuáles son las principales contribuciones de las siguientes personas: Stephen Cook, Richard Karp, Alan Turing, Alonzo Church, Claude Shannon, Donald Knuth, Andrei Kolmogorov, John Von Neumann, Emil Post, Manindra Agrawal, Leonid Khachiyan?

II Clases de Complejidad y Modelos No Deterministas

1. Falso o verdadero:
   (a) $\mathrm{P} \subseteq \mathrm{NP}$.
   (b) $\mathrm{P} \cap \mathrm{NP} = \varnothing$.
   (c) P es un conjunto finito.
   (d) NP es un conjunto finito.
   (e) NP es un conjunto de algoritmos.
   (f) NP es un conjunto de problemas.
   (g) NP es un problema.
   (h) $\mathrm{NPC} \cup \mathrm{P} = \mathrm{NP}$.
   (i) Los problemas NP-completos son los más difíciles de todos.

(j) El problema de *encontrar* ciclos Hamiltonianos es NP-completo.
(k) El problema de *encontrar* ciclos Hamiltonianos es NP-difícil.
(l) El problema del paro es NP-difícil.
(m) NP significa “No polinomial”.
(n) Nadie conoce ningún algoritmo polinomial para ningún problema en NP.
(ñ) Los problemas NP-completos son no tratables.
(o) Los problemas NP-completos no tienen (ni van a tener nunca) algoritmos polinomiales que los resuelvan.
(p) P = NP $\cap$ co-NP.
(q) P = co-P.
(r) ¿Qué es un *algoritmo* NP?
(s) Si P = NP entonces todos los problemas son NP-difíciles.

2. Suponga que finalmente, la Humanidad logra construir computadoras cuánticas y que, como se espera, esto permita resolver todos los problemas NP en tiempo polinomial. ¿Esto resuelve el problema P $\stackrel{?}{=}$ NP? Explique.

3. ¿Usted se inclina más a creer que P = NP o que P $\neq$ NP? ¿Por qué? ¿podría ser más bien indecidible? ¿qué significaría eso?

4. Describa en detalle cada uno de los siguientes modelos:

   (a) Máquina de Turing No Determinista.
   (b) Máquina de Turing “con oráculo”
   (c) RAM de costo logarítmico con “choice”

5. Defina: Certificado Sucinto, Problema Polinomialmente Verificable.

6. Muestre que todo problema en NP puede resolverse en una MT *determinista* en tiempo cuando mucho $T(n) = O(2^{p(n)})$, para algún polinomio $p(n)$.

7. Muestre que cualquier problema en NP puede resolverse en tiempo polinomial por una RAM de costo uniforme que incluya, además de las operaciones ordinarias, operaciones booleanas bit-a-bit entre las representaciones binarias de enteros arbitrariamente grandes.

8. Defina: Transformación Polinomial (de Karp), Reducción Polinomial (de Cook).
9. Defina: NP-completo, NP-Difícil, NP, P, co-NP, co-NP-completo, Problema no tratable.
10. Demuestre que $\Pi_1 \rightsquigarrow \Pi_2 \rightsquigarrow \Pi_3$ implica $\Pi_1 \rightsquigarrow \Pi_3$.
11. Demuestre que $\Pi_1 \rightsquigarrow \Pi_2 \in \mathrm{P}$ implica $\Pi_1 \in P$.
12. Demuestre que $\Pi_1 \rightsquigarrow \Pi_2$, $\Pi_1 \in \mathrm{NPC}$ y $\Pi_2 \in \mathrm{NP}$ implican que $\Pi_2 \in \mathrm{NPC}$.
13. Demuestre que son equivalentes:
    (a) $\mathrm{P} = \mathrm{NP}$.
    (b) SAT tiene un algoritmo polinomial que lo resuelve.
    (c) $\mathrm{P} \cap \mathrm{NPC} \neq \varnothing$.
    (d) Todo problema es NP-difícil.
14. El teorema de Cook demuestra que siempre *existe* una transformación polinomial de cualquier problema dado $\Pi \in \mathrm{NP}$ a SAT. ¿Puede esta transformación ser efectivamente programada?

III Problemas NP-completos.

1. De un esbozo del teorema de Cook.
2. Muestre que 3SAT es NP-completo.
3. Muestre que VertexCover es NP-completo.
4. Muestre que IndependentSet es NP-completo.
5. Muestre que Clique es NP-completo.
6. Muestre que DirectedHamiltonianCycle es NP-completo.
7. Muestre que HamiltonianCycle es NP-completo.
8. Muestre que TravelingSalesman Problem es NP-completo.
9. Muestre que 3DimensionalMatching es NP-completo.
10. Muestre que Partition es NP-completo.
11. Muestre que el problema de la Mochila es NP-completo.
12. Resuelva todos los ejercicios propuestos al final del capítulo 3 del libro de Garey y Johnson.