MODELOS AVANZADOS DE COMPUTACIÓN

Relación 5

Clases de Complejidad

1. (a) Un grafo dirigido se dice que es *acíclico* si no tiene ciclos. Demostrar que todo grafo dirigido acícilo tiene una *fuente* (un nodo al que no llegan arcos).

   (b) Demostrar que un grafo dirigido con n nodos es acíclico si y solo si se pueden numerar los nodos del 1 al n de manera que siempre los arcos van desde números más pequeños a números más grandes.

   (c) Describir un algoritmo polinómico para determinar cuando un grafo es acíclico.

2. (a) Demostrar que un grafo es *bipartito* (se puede dividir en dos partes, que no son necesariamente iguales, de manera que todos las aristas van de una parte a otra) si y solo si todos sus ciclos son de longitud par.

   (b) Describir un algoritmo polinómico para comprobar si un grafo es bipartito.

3. Demostrar que $\mathbf{P}$ es cerrada para la unión y la intersección.

4. Demostrar que $\mathbf{NP} \neq \mathbf{ESPACIO}(n)$

5. Sea C una clase de funciones de los enteros no negativos en los enteros no negativos. Se dice que C es cerrada para la composición polinómica por la izquierda si $f(n) \in C$ implica que $p(f(n)) = O(g(n))$ para alguna función $g(n) \in C$, y donde $p(n)$ es un polinomio cualquiera. Se dice que C es cerrada para la composición polinómica por la derecha si $f(n) \in C$ implica que $f(p(n)) = O(g(n))$ para alguna función $g(n) \in C$, y donde $p(n)$ es un polinomio cualquiera.

   Determinar cuales de las siguientes clases de funciones son cerradas para la composición polínómica por la derecha y cuales lo son por la izquierda.

   (a) $\{n^k \,:\, k > 0\}$

   (b) $\{n.k \,:\, k > 0\}$

   (c) $\{k^n \,:\, k > 0\}$

   (d) $\{2^{n^k} \,:\, k > 0\}$

   (e) $\{\log^k(n) \,:\, k > 0\}$

   (f) $\{\log(n)\}$

Nota: en los dos últimos casos se entiende que las funciones se redondean al entero más cercano. En vez de considerar todos los polinomios, como estamos hablando de órdenes de complejidad será suficiente considerar los polinomios de la forma $p(n) = n^l$, donde l es un número natural.

6. Sea $L = \{wcw : w \in \{0,1\}^*\}$,
   - Demostrar que L se puede reconocer en espacio $\log(n)$.
7. Demostrar que si L esta en **P**, entonces L^* también está en **P**.
8. Demostrar que si L esta en **NP**, entonces L^* también está en **NP**.
9. Demostrar que **NP** es cerrada para la unión y la intersección.
10. Demostrar que si **NP** $\neq$ **CoNP** entonces **P** $\neq$ **NP**
11. Supongamos que una entrada es una palabra de paréntesis. Demostrar que determinar si están emparejados y anidados correctamente está en **L**. Lo están (()()) y ((())), pero no lo están)()(ni (()))(.
12. Consideremos el problema anterior, pero ahora con dos tipos de paréntesis (,) y [,]. Demostrar que determinar si están bien escritos también está en **L**. Aquí ([]()) está permitido, pero ([)] no. Demostrar que si la entrada se lee de izquierda a derecha sin volver hacia atrás, requieren $O(n)$ de memoria.
13. Demostar que el problema del palíndromo, requiere $O(n)$ de memoria si no puede volver hacia atrás.