Tecnología Digital V: Diseño de algoritmos
Licenciatura en Tecnología Digital
Primer Semestre 2025

Guía de ejercicios 5: Teoría de NP-completitud

1. Mostrar que los siguientes problemas de decisión pertenecen a la clase P.
   a) Determinar si una lista de n números está ordenada.
   b) Chequear si un grafo es conexo.
   c) Chequear si un grafo tiene un vértice de grado 7 o más.
2. Mostrar que los siguientes problemas pertenecen a NP, diseñando certificados polinomiales. Para cada uno, describir el algoritmo polinomial que permite validar que el certificado es correcto.
   a) Determinar si un grafo tiene más de una componente conexa.
   b) Determinar si dos grafos son isomorfos.
   c) Determinar si un grafo tiene un ciclo.
   d) Determinar si un grafo tiene un circuito hamiltoniano (es decir, un ciclo que pase por todos los vértices).
   e) Determinar si un grafo es 5-coloreable.
3. Considerar el conjunto $\mathcal{G}$ de todos los grafos que contienen a K_4 como subgrafo inducido (es decir, que contienen cuatro vértices vecinos dos a dos). ¿Pertenece a P el problema de determinar si un grafo pertenece a $\mathcal{G}$? ¿Pertenece a NP?
4. Mostrar que la relación $\leq_p$ es transitiva (es decir, si $P_1 \leq_p P_2$ y $P_2 \leq_p P_3$ entonces $P_1 \leq_p P_3$).
5. Decir si las siguientes afirmaciones son verdaderas, falsas, o si no se sabe:
   a) $P \subseteq NP$
   b) Si $P = NP$ entonces todo problema NP-completo está en P
   c) Si un problema A es NP-completo y $A \in P$, entonces $P = NP$
6. ¿Es cierto que si dos problemas X e Y son NP-completos entonces $X \leq_p Y$, y también $Y \leq_p X$? Justificar la respuesta.
7. Sean X e Y dos problemas de decisión tales que $X \leq_p Y$. ¿Cuáles de las siguientes afirmaciones son correctas?
   a) Si X está en *P* entonces Y está en *P*.
   b) Si Y está en *P* entonces X está en *P*.
   c) Si Y es *NP-completo* entonces X también.
   d) Si X es *NP-completo* entonces Y también.
   e) Si Y es *NP-completo* y X está en *NP* entonces X es *NP-completo*.
   f) Si X es *NP-completo* e Y está en *NP* entonces Y es *NP-completo*.
8. Decir si las siguientes afirmaciones son verdaderas o falsas:
   a) Si $P = NP$ entonces todo problema NP-completo es polinomial.
   b) Si $P = NP$ entonces todo problema NP-hard es polinomial.

9. El problema k-COLOREO pregunta si un grafo $G = (V, E)$ puede ser coloreado con k colores, de modo que no haya dos vértices adyacentes que tengan el mismo color.
   a) Supongamos que el problema 4-COLOREO es NP-completo. Demostrar que entonces 5-COLOREO también es NP-completo.
   b) ¿Se puede usar el mismo argumento para demostrar que 3-COLOREO es NP-completo? ¿Por qué?
10. Un conjunto independiente en un grafo $G = (V, E)$ es un subconjunto de vértices tal que no hay aristas entre ellos. Sea $\bar{G}$ el complemento de G.
    a) Demostrar que S es una clique de tamaño k en G si y solo si S es un conjunto independiente de tamaño k en $\bar{G}$.
    b) Usar este resultado para demostrar que si CLIQUE es NP-completo, entonces CONJUNTO-INDEPENDIENTE también es NP-completo.
11. El problema PARTITION pregunta si un conjunto de enteros positivos se puede dividir en dos subconjuntos con la misma suma.
    a) Mostrar que PARTITION $\in$ NP.
    b) Dada una instancia (S, k) de SUBSET-SUM donde $S = \{a_1, a_2, \ldots, a_n\}$ y la suma total es T, construir una instancia de PARTITION agregando dos números nuevos al conjunto S. Pista: agregar $(T - k)$ y T.
    c) Verificar que la transformación funciona correctamente.
12. ¿Qué se puede inferir del hecho de que el problema del viajante de comercio (TSP) es NP-completo, suponiendo $P \neq NP$?
    a) No existe un algoritmo que resuelva instancias arbitrarias de TSP.
    b) No existe un algoritmo que eficientemente resuelva instancias arbitrarias de TSP.
    c) Existe un algoritmo que eficientemente resuelve instancias arbitrarias de TSP, pero nadie lo ha encontrado.
    d) TSP no está en P.
    e) Todos los algoritmos que resuelven TSP corren un tiempo polinomial para algunas entradas.
    f) Todos los algoritmos que resuelven TSP corren un tiempo exponencial para todas las entradas.
13. ¿Qué se puede inferir del hecho de que el problema de determinar si dos grafos son isomorfos está en NP pero no se sabe si es NP-completo, suponiendo $P \neq NP$?
    - Existe un algoritmo que resuelve instancias arbitrarias de *Isomorfismo*.
    - Existe un algoritmo que resuelve eficientemente instancias arbitrarias de *Isomorfismo*.
    - Si encontramos un algoritmo polinomial que resuelva *Isomorfismo*, podemos usarlo como una "caja negra" para resolver TSP.
14. Discutir algunas de las alternativas posibles cuando se debe resolver en la práctica un problema que se sabe que es NP-completo.
15. ¿Cuál sería la importancia de que alguien encuentre un algoritmo polinomial que resuelve un problema NP-completo? ¿Cuál sería la importancia de que alguien pruebe que un problema en NP es intratable?
16. ¿Cuál es la diferencia entre un algoritmo polinomial y un algoritmo pseudo-polinomial? Dar un ejemplo de un algoritmo pseudo-polinomial. Indicar en qué circunstancias este algoritmo presenta dificultades.