

1. ¿Cuál es la definición formal de NP? ¿Cómo se relaciona con la clase de problemas P?

Los problemas NP se pueden clasificar como problemas que se pueden resolver en un tiempo polinomial no determinístico. La relación con los problemas P es que estos sí se pueden resolver en un tiempo polinomial determinístico.

2. ¿Cuál es la definición formal de NPC? ¿Por qué se dice que los problemas en NPC son los problemas más difíciles en NP?

La definición de los problemas NPC son los problemas más difíciles que existen en NP, es decir que su solución no puede ser encontrada incluso en un tiempo polinomial no determinístico. Dentro de los problemas NP es difícil encontrar la solución y fácil de comprobar mientras que en los problemas NPC esta solución y demostración es sumamente difícil.

3. Dado un problema de decisión, ¿cómo podemos determinar si pertenece a NP? Proporcione un ejemplo.

Un ejemplo sería un rompecabezas, posiblemente hacer un algoritmo que pruebe todas las posibles combinaciones de las piezas de un rompecabezas sería sumamente difícil hacerlo, pero la comprobación sería sumamente sencilla ya que basta con verlo y ver si la imagen del rompecabezas es la correcta.

4. Proporcione un ejemplo de un problema en NPC y explique por qué es difícil resolverlo.

Un ejemplo de problema en NP completo es el "Problema de la Mochila", que implica maximizar el valor total de elementos que pueden caber en una mochila limitada por capacidad. Aunque verificar soluciones propuestas es fácil, encontrar la solución óptima requiere evaluar todas las combinaciones posibles, lo que lleva a un tiempo de ejecución exponencial, esto incrementaría en función de los elementos disponibles para introducir en la mochila. Esto lo convierte en un problema difícil de resolver y clasifica como NPC.

5. ¿Cómo podemos demostrar que un problema es NP-completo? Explique el proceso de reducción.

Según mi conocimiento el proceso de reducción es encontrar una forma de reducir el problema, un es con el problema de mochila. Esto es, podríamos reducir los elementos disponibles para meter dentro de la mochila y obtener una solución y así el problema sería NP y podríamos decir que está reducido. Es decir, se debe mostrar que, para cualquier instancia del problema, existe un algoritmo que puede verificar una solución propuesta en tiempo polinomial.

6. ¿Qué significa que un problema sea NP-duro? ¿Cómo se relaciona con NPC?

No tengo nociones de que sea un NP duro.

7. ¿Cómo podemos demostrar que un problema es NP-completo utilizando el problema de satisfacción de restricciones (CSP, por sus siglas en inglés)?

No tengo noción de esta pregunta.

8. ¿Cómo se relaciona la existencia de un algoritmo eficiente para resolver un problema NP-completo con la clase de problemas P?

La existencia de un algoritmo eficiente para resolver un problema NP-completo implicaría que la clase de problemas NP sería igual a la clase de problemas P. En otras palabras, si un problema NP-completo puede resolverse en tiempo polinomial, entonces todos los problemas en NP podrían resolverse en tiempo polinomial, lo que implica que $P = NP$. Esto tendría enormes implicaciones en la teoría y la práctica de la computación, ya que problemas difíciles podrían resolverse eficientemente, pero también podría comprometer la seguridad de sistemas criptográficos.

9. Proporcione un ejemplo de un problema en NP que no se ha demostrado que sea NPC.

He oído de "Problema del Emparejamiento Máximo" (Maximum Matching Problem).

En este problema, se busca encontrar el conjunto más grande de aristas no adyacentes en un grafo bipartito. Es decir, se busca encontrar el mayor subconjunto de aristas tal que ningún par de aristas comparta un vértice extremo.

10. ¿Cuál es la importancia de la pregunta P vs. NP en la teoría de la complejidad computacional? ¿Cuáles son las implicaciones prácticas si se encuentra una respuesta afirmativa o negativa?

En base a lo que se ha visto en clase, la pregunta P vs. NP es uno de los problemas más importantes en la teoría computacional. Si se demuestra que $P = NP$, significaría que cualquier problema cuya solución pueda ser verificada en tiempo polinomial también puede ser resuelto en tiempo polinomial. Esto tendría un impacto revolucionario en diversos campos, ya que problemas difíciles, como los relacionados con la criptografía y la inteligencia artificial, podrían ser resueltos de manera eficiente. Por otro lado, si se prueba que $P \neq NP$, significaría que existen problemas cuya solución es difícil de encontrar, incluso si es fácil verificarla. Esto reforzaría la seguridad en criptografía, ya que los sistemas basados en la dificultad de resolver problemas NP-completos seguirían siendo seguros. Sin embargo, problemas complejos seguirían siendo difíciles de resolver rápidamente, lo que podría limitar el avance en áreas como la optimización y la inteligencia artificial.