Instituto Politécnico Nacional

Escuela Superior de Computo

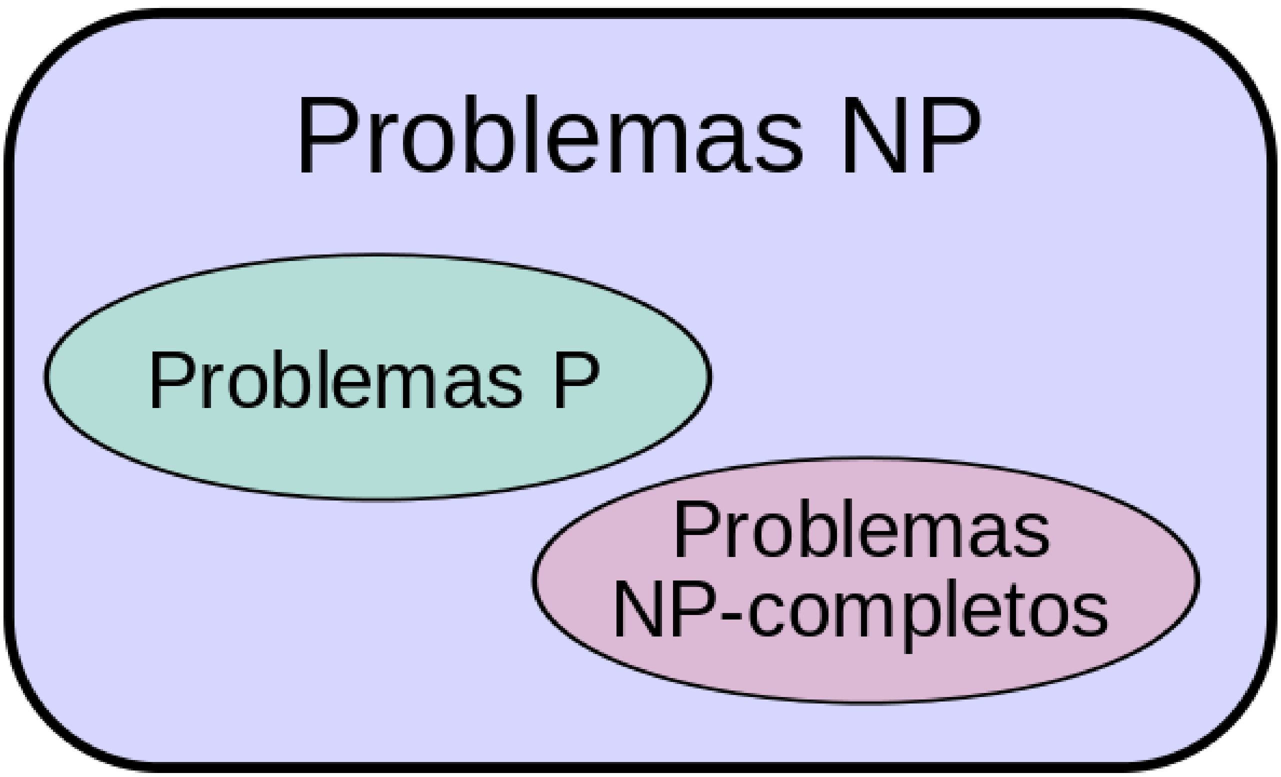
Analisis y Diseño de Algoritmos

Tarea: complejidad

Alumno:

Rivera Plascencia Bryan

Grupo: 3BV1



Complejidad P: problema resolubles en tiempo polinomico

Existe un algoritmo de complejidad polinómica $O(n^k)$ que resuelve el problema (k constante, n tamaño de la entrada del problema)

En computación, cuando el tiempo de ejecución de un algoritmo es menor o igual que un cierto valor calculado a partir del numero de variables implicadas usando una formula polinómica, se dice que dicho problema se puede resolver en un tiempo polinómico o polinomial P.

La clase de complejidad P consiste de aquellos problemas algorítmicos uqe podemos resolver en tiempo polinomial. Con esto nos referimos a que para ellos debe existir un entero k >= 0 y un algoritmo tal que cualquier instancia de tamaño n la resuelva en tiempo $O(n^k)$.

Ejemplos:

- Hallar el MTS de un grafo con pesos en las aristas.
- Hallar el camino de peso minimo entre dos vértices de un grafo.
- Decidir si un grafo es euleriano.

Complejidad NP: Problemas verificables en tiempo polinomico

Dado un “ceriftificado” (una posible solucion), podemos verificar si es correcto en tiempo polinómico en el tamaño del problema

La importancia de esta clase de problemas de decisión es que contiene muchos problemas de búsqueda y de optimización para los que se desea saber si existe una cierta solución o si existe una mejor solución que las conocidas. En esta clase están el problema del viajante donde se quiere saber si existe una ruta optima que pasa por todos los nodos de un cierto grafo, donde se desea saber si una cierta formula de lógica proposicional pode ser cierta para algún conjunto de valores booleanos para las variables.

Son aquellos problemas cuya respuesta “si” se puede verificar en tiempo polinomial. Siendo un poco más precisos, lo que esperamos de los problemas en NP es que cada que la respuesta sea “si” haya un objeto testigo que testifique que la respuesta sea “si” y un **algoritmo verificador** que verifique la correctitud del testigo en tiempo polinomial.

Ejemplos:

- Decidir si un grafo es 4-coloreable
- Decidir si un grafo es hamiltoniano
- Hallar el camino mas largo entre dos vértices del grafo

Complejidad de NP-completo

Es el subconjunto de los problemas más difíciles de NP se puede reducir en cada uno de los problemas de NP-completo. Se puede decir que los problemas de NP-completo son los problemas mas difíciles de NP y muy probable no formen parte de la clase de complejidad P.

La razón es que de tenerse una solucion polinómica para un problema NP-completo, todos los problemas de NP-completo, completo, todos los problemas de NP tendrían tambien una solucion en tiempo polinomico.

Si se demostrase que un problema NP-completo, llamémosla A, no se pudiese en tiempo polinomico, el resto de los problemas NP-completos tampoco se podrían resolver en tiempo polinomico. Esto se debe a que si uno de los problemas NP-completos distintos de A, digamos X, se pudiese resolver en tiempo polinómico, entonces A se podría resolver en tiempo polinomico, por definición de NP-completo. Ahora, pueden existir problemas en NP y que no sean NP-completos para los cuales exista solución polinómica, aun no existiendo solucion para A.

Ejemplo:

Encontramos el problema de subconjuntos que se puede enunciar como sigue: dado un conjunto S de enteros, ¿existe un sobconjunto no vacio de S cuyos elementos sumen cero? Es fácil verificar si una respuesta es correcta, pero no se conoce mejor solución que explorar todos los $2^n – 1$ subconjuntos posibles hasta encontrar uno que cumpla con la condición.

Complejidad NP-Dificil

Quiere decir al menos tan complejo como NP(pero no necesariamente NP).

Es el conjunto de los problemas de decisión que contiene los problemas H tales que que todo problema L en NP puede ser transformado polinomialmente en H.

Esta clase puede ser descrita como aquella que contiene a los problemas de decisión que son como minimo tan difíciles como un problema de NP.

Esta afirmación se justifica porque si podemos encontrar un algoritmo A que resuelve uno de los problemas H de NP-dificil en tiempo polinomico, entonces es posible construir un algoritmo que trabaje en tiempo polinomico para cualquier problema NP ejecutando primero la reducción de este problema en H y luego ejecutando el algoritmo A.

Bibliografía:

https://es.wikipedia.org/wiki/P_(clase_de_complejidad)

https://es.wikipedia.org/wiki/NP-completo#:~:text=En%20teor%C3%ADa%20de%20la%20complejidad,los%20problemas%20de%20NP%2Dcompleto.

http://madi.nekomath.com/P5/ReduccionesPvsNP.html

https://blogs.upm.es/gregoriohp/wp-content/uploads/sites/1071/2023/01/ComplejidadNP.pdf

https://www.wikiwand.com/es/NP-completo