UNIVERSIDAD NACIONAL

AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

PROBLEMAS P Y NP

Adrián Ulises Mercado Martínez

PACHECO SALGADO MAURICIO

Problemas P

Algoritmo de Dijkstra

También llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto del vértices en un grafo con pesos en cada arista.

Torre de Hanoi

Las Torres de Hanoi es un juego de lógica donde tenemos tres torres y tenemos que mover los discos que hay en la primera torre a la tercera, siguiendo siempre algunas reglas, como que sólo podemos mover los discos de uno en uno o que no podemos poner un disco sobre otro más pequeño que él.

La teoría de la complejidad computacional o teoría de la complejidad informática es una rama de la teoría de la computación que se centra en la clasificación de los problemas computacionales de acuerdo con su dificultad inherente, y en la relación entre dichas clases de complejidad.

Un problema se cataloga como difícil si su solución requiere de una cantidad significativa de recursos computacionales, sin importar el algoritmo utilizado. La teoría de la complejidad computacional formaliza dicha aseveración, introduciendo modelos de computación matemáticos para el estudio de estos problemas y la cuantificación de la cantidad de recursos necesarios para resolverlos, como tiempo y memoria.

La relación entre las clases de complejidad P y NP es estudiada por la teoría de la complejidad computacional, la parte de la teoría de la computación que trata de los recursos requeridos durante el cálculo para resolver un problema dado. Los recursos más usuales son tiempo (¿cuántos pasos son necesarios para resolver un problema?) y espacio (¿cuánta memoria es necesaria para resolver un problema?).

En esta teoría, la clase P consiste de todos aquellos problemas de decisión que pueden ser resueltos en una máquina determinista secuencial en un período de tiempo polinomial en proporción a los datos de entrada. En la teoría de complejidad computacional, la clase P es una de las más importantes; la clase NP consiste de todos aquellos problemas de decisión cuyas soluciones positivas/afirmativas pueden ser verificadas en tiempo polinómico a partir de ser alimentadas con la información apropiada, o en forma equivalente, cuya solución puede ser hallada en tiempo polinómico en una máquina no determinista.

Más precisamente, un *problema de decisión* es un problema que especifica una cadena de caracteres de datos de entrada y requiere como solución una respuesta por el SI o por el NO.

En forma intuitiva, consideramos que los problemas contenidos en P son aquellos que pueden ser resueltos en forma razonablemente rápida.

Dentro de estos “problemas difíciles” hay varias subdivisiones, pero hoy sólo mencionaremos la clase NP. Serían aquello problemas que a día de hoy no podemos resolver (porque no hemos encontrado aún un algoritmo, etc., etc., etc.), pero para los que, conocida una solución

particular, podríamos verificar que es correcta, que resuelve el problema, en tiempo polinómico.

Problemas NP

Problemas de satisfacibilidad booleana

En 1960 Martin Davis y Hilary Putnam desarrollaron un algoritmo para comprobar la satisfacibilidad de las fórmulas de la lógica proposicional en FNC; es decir, en un conjunto de cláusulas unidas por conjunciones. El algoritmo usa una forma de resolución en la cual las variables son elegidas iterativamente y eliminadas mediante la resolución de cada cláusula donde la variable aparezca afirmada con una cláusula en la que la variable esté negada. En 1962 se desarrolló el algoritmo DPLL por Davis-Putnam-Logemann-Lovelandes, un algoritmo completo basado en la vuelta atrás que sirve para decidir la satisfacibilidad de las fórmulas de lógica proposicional en una forma normal conjuntiva; es decir, para resolver el problema FNC-SAT, que al igual que hacía el algoritmo anterior de Davis y Putnam.

Problema de optimización de cobertura de conjuntos

En el problema de optimización de cobertura de conjuntos, la entrada es un par {\displaystyle ({\mathcal {U}},{\mathcal {S}})} y la tarea es encontrar un conjunto de cobertura que use los menos conjuntos posibles.