



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Adrian Ulises Mercado

Profesor:

Asignatura:

Estructura de datos y Algoritmos
I

Grupo:

13

No de Práctica(s):

Tarea 3 NP Y P

Integrante(s):

Martínez Jacques Ricardo

No. de Equipo de cómputo empleado:

No. de Lista o Brigada:

BRIGADA 5

Semestre:

2020-2

Fecha de entrega:

07-06-2020

Observaciones:

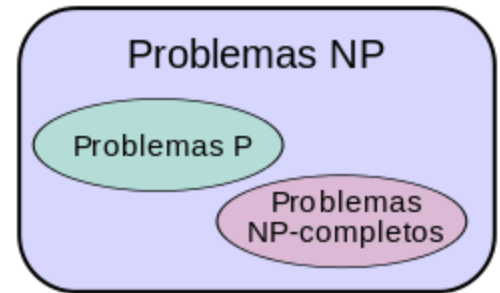
CALIFICACIÓN: _____

LOS PROBLEMAS de complejidad TIPO NP Y P

En teoría de la complejidad computacional, una clase de complejidad es un conjunto de problemas de decisión de complejidad relacionada.

Una clase de complejidad tiene una definición de la forma:

- el conjunto de los problemas de decisión que pueden ser resueltos por una máquina M utilizando $O(f(n))$ del recurso R (donde n es el tamaño de la entrada).



La relación entre las clases de complejidad NP y P es una pregunta por primera vez formulada por el científico computacional Stephen Cook que la teoría de la complejidad computacional aún no ha podido responder.

Supongamos el problema de sacar la raíz de un número y elevar un número para sacar la raíz, resulta más sencillo elevar el número que sacar la raíz, ya que pocos saben cómo desarrollar un raíz.

La complejidad P.

P es la clase de complejidad que contiene problemas de decisión que se pueden resolver en un tiempo polinómico.

P contiene a la mayoría de los problemas naturales, algoritmos de programación lineal, funciones simples y más.

Por ejemplo, la suma de dos números naturales se resuelve en tiempo polinómico (ósea orden $2n$). Entre los problemas que se pueden resolver en tiempo polinómico nos encontramos con diversas variedades como los logarítmicos ($\log(n)$), los lineales (n), los cuadráticos (n^2), los cúbicos (n^3), etc.

Volviendo al ejemplo principal llegamos a la conclusión que la función de elevar al cuadrado está contenida en la clase P.

La complejidad NP

La clase de complejidad NP contiene problemas que no pueden resolverse en un tiempo polinómico. Cuando se dice que un algoritmo no puede obtener una solución a un problema en tiempo polinómico siempre se intenta buscar otro procedimiento que lo consiga mejorar. Este tipo de complejidad tiene un método más complejo que el del tipo P.

Podemos ver que la operación de calcular la raíz cuadrada se encuentra contenida en esta clase.

Y se puede decir que la clase de complejidad es un subconjunto del tipo NP.

Ya que si se requiere comprobar la solución de la raíz de un numero (NP), basta con elevar este mismo al índice de la raíz (P).

Pero la cuestión por la que aun no se ha podido resolver es si se puede hacer lo inverso. Y saber si

¿todo problema puede resolverse en un tiempo polinómico?