ESTRUCTURA DE DATOS Y ALGORITMOS I **2020-2**

Revisión de ejercicios - Complejidad de códigos y P VS NP

- **Preguntas frecuentes y más relevantes**

¿Cómo afecta en el análisis de complejidad una condicional if?

Por sí misma la complejidad de la estructura if siempre es de orden constante debido a que no se relaciona con la cantidad de datos o el tamaño de la entrada, ya que el funcionamiento de un if es a través de una, dos, o más comparaciones. Pero estas implican un número constante de operaciones.
Lo que sí puede ocurrir es lo que vimos en el ejemplo del video; que la condicional pueda tener dos categorías de complejidad diferentes (una para el if, otra para else, etc.) y para estos casos en el análisis de complejidad se toma como el más grande.

En una estructura switch-case ¿el análisis es similar al de un if?

Totalmente, de la misma manera que en el if, la estructura por sí misma tiene orden de complejidad constante, ya que sólo realiza la evaluación de un valor para saber a qué caso entrar y realizar esas instrucciones.
También puede ocurrir que cada uno de los casos tenga una complejidad diferente.

¿Qué es más eficiente, una función recursiva lineal o una iterativa lineal de la misma complejidad?

En principio, son igual de eficientes, posiblemente un análisis a posteriori con muchos datos nos podría dar ligeramente una de las dos como más eficiente.
Recuerden que la recursividad no es buena o mala por sí misma; depende del algoritmo que se analiza.
Lo que sí puede pasar es que haya para un mismo problema una solución recursiva y una solución "iterativa" y que las soluciones puedan tener diferentes complejidades.

¿Es posible expresar cualquier función recursiva de forma iterativa?

Esto es un spoiler de la siguiente clase virtual, pero sí es posible.

Si dos algoritmos tienen el mismo orden de complejidad, pero distinta cantidad de instrucciones simples, ¿es más complejo el que tiene más instrucciones?

En este caso sí porque la cantidad de instrucciones simples representa la constante c.
Si tenemos 65n y 123n sería de mayor orden 123n

En el campo laboral, desarrollo de videojuegos, ciencia de datos, investigación ¿se realizan análisis de complejidad?

En empresas medianas y pequeñas no, porque las prioridades son otras, pero en empresas grandes y trasnacionales tienen departamentos de investigación especializados si se hacen análisis de complejidad para tomar decisiones en cuanto a programas especializados que se realizan.
En áreas de reciente desarrollo como la de ciencias de datos me parece que los análisis de complejidad son fundamentales para entender mejor o buscar una mayor eficiencia en los procesos de análisis de datos.
Por cierto ¿alguien está interesado en cambiarse a la carrera de ciencias de datos?

¿El uso de los ciclos while implica una complejidad mayor que un for?

No, el inicio o fin de ciclo puede realizarse a través de la evaluación de una dos o algunas operaciones lo que representa orden constante, lo que daría una complejidad diferente sería si dentro del while el crecimiento de las operaciones tiende a n, n^2, n^3

¿Influyen los tipos de datos (float, char etc.) en la complejidad?

No influyen porque lo que varía de uno a otro es la forma en que utiliza la memoria y la forma en la que se interpreta el patrón de bits.

¿Cómo afecta el uso de funciones (especialmente en funciones anidadas) en un análisis de complejidad?

En principio no afectan el resultado final pero sí suele ser más complicado realizar el análisis porque se tiene que ver hasta dónde están anidadas las funciones y a partir de ahí contabilizar el número de operaciones (como una especie de análisis bottom-up) pero normalmente el uso de funciones se establece más al momento de programar el algoritmo para hacerlo más modular, fácil de verificar y revisar.
Los algoritmos teóricos por sí mismos no siempre implican una descomposición en funciones.

¿Qué pasa cuando en un algoritmo el tamaño de la entrada se establece en la combinación de dos o más variables (n, m)?

En estos casos el análisis de complejidad resulta más complicado porque se tiene que considerar como es el crecimiento de ambas variables, si una está en función de otra pues a final de cuentas la complejidad la determina la que crece más, Pero si son variables que están por separado, se complica.
No sé si alguien en la tarea puso algún ejemplo de un análisis de complejidad de este tipo.
--

Hubo otras preguntas que se repitieron bastante pero que se respondieron (espero) al proporcionarles la solución de la complejidad de los códigos como por ejemplo, si la "i" (de un for) no se incrementa de uno en uno puede cambiar la complejidad, o también el hecho de que en las funciones recursivas si la "n" no se resta de uno en uno también puede cambiar su complejidad.

Al buscar en Internet, algunos problemas aparecen algunas veces aparecen como NP y en otros lados aparecen como NP-Completos

Tengan cuidado con las fuentes que revisan, en varios lugares suele haber confusión respecto a la forma en la que se explican las categorías de complejidad.
Sin embargo, es importante señalar que la categoría P o NP muchas veces se aplica en problemas de decisión (si o no) y algunos problemas pueden tener variantes en la decisión que se evalúa y dependiendo de esta podrían aparecer como NP o NP completos.

Existe algún método para encontrar soluciones aproximadas a problemas no factibles que sea el mejor. Y cuál es el método más utilizado en la vida real

Como siempre, depende del tipo de problema que se resuelve, aplicar la estrategia más adecuada. En algunos casos conviene la aproximación en otros los métodos probabilísticos. En mi opinión el más utilizado es el de restricciones ya que se trata de soluciones que son eficientes y que son para conjuntos de datos limitados.

¿Por qué si aún no se demuestra si son iguales o diferentes se representa a P como un subconjunto de NP?

Se representan así porque como hasta el momento no existe una solución eficiente para los problemas NP se asume que P != NP
Sin embargo, como esta afirmación no ha sido demostrada existe la esperanza de encontrar ese algoritmo eficiente.

¿En qué clase qué agrupan los algoritmos de inteligencia artificial?

Suelen ser NP completo porque involucran una serie de algoritmos para el aprendizaje autónomo y la verificación de todo este tipo de soluciones suele ser NPC.

¿Qué diferencia hay entre la máquina de Turing y la computación cuántica?

Máquina de Turing es un modelo matemático y la computación cuántica está basada en la física cuántica, la cual es una disciplina diferente.

¿Hasta qué punto o en qué criterio se considera algo como "sencillo o no sencillo" de verificar?

Esta pregunta es muy buena y difícil de explicar, pero tiene que ver con la dificultad de verificar la solución si el "tamaño" del problema aumenta.
Si al aumentar el tamaño la solución se vuelve mas difícil de verificar, es NP-Completo

¿Por qué una computadora cuántica podría reducir la complejidad de los problemas?

En realidad, no se sabe si esto pueda pasar, lo que se tendría es una nueva alternativa, ya que los algoritmos de una computadora cuántica funcionarían de manera diferente que la computación tradicional y al funcionar de manera diferente tal vez podrían mostrar una solución.

Hubo varias preguntas de acerca de las implicaciones sobre qué pasa si son iguales son diferentes, precisamente la explicación de estas implicaciones está en el artículo que están leyendo aunque quizá lo complejo es entender el porqué de esas implicaciones; una manera burda de explicarlo es que si se encuentra un algoritmo eficiente cualquier problema sería "fácil" de resolver .

- **Ejemplos que ustedes propusieron en las categorías NP Y NPC**

Aquí es importante aclarar que la respuesta correcta sobre la clase a la que pertenecen los problemas que se muestran, depende de la demostración matemática correspondiente relacionada con algún problema NP, en particular el de la Satisfacibilidad Booleana.

NP

- ✓ Encontrar la llave correcta para abrir la puerta en un llavero
- ✓ Encontrar una contraseña
- ✓ Figura de origami
- ✓ Un examen de opción múltiple
- ✓ Realizar una receta de cocina

NP-Completo

- ✓ Sopa de letras
- ✓ Buscaminas (Es posible encontrar una solución a partir de un punto de inicio)
- ✓ Super Mario Bros (ver página)
- ✓ Ajedrez (casi en cualquier enfoque del juego)
- ✓ El problema de la mochila
- ✓ Problema de las n reinas
- ✓ Cuadrado mágico

No estoy seguro xd

- ✓ Crucigramas
- ✓ Sincronización de semáforos
- ✓ El juego de Battleship