

Introducción

El objetivo de este proyecto de investigación y análisis es evaluar la comprensión del alumno o alumna sobre la relación entre las técnicas de programación dinámica y *Divide and Conquer* (DaC), así como su aplicabilidad y sus resultados en términos de desempeño, en la resolución de un problema computacional.

Las condiciones son las siguientes:

- Fecha de entrega: semana del 27 al 28 de abril de 2023.
- Cantidad máxima de personas en un grupo de trabajo: 3.
- Modalidad de entrega: virtual.
- Cada grupo debe trabajar sobre un problema diferente.
- No es permitido trabajar sobre problemas vistos en clase.

Instrucciones

Usted (y su grupo, si trabaja con más personas) deberá investigar un único problema computacional y proporcionar dos algoritmos para resolverlo. Uno de los algoritmos debe estar desarrollado con la técnica *Divide and Conquer* y el otro con programación dinámica. Ambos algoritmos deben ser analizados por conteo de operaciones usando las metodologías de análisis vistas en clase para sus correspondientes acercamientos de diseño. Además, ambos algoritmos deben ser programados y analizados de forma empírica, produciéndose una gráfica por algoritmo que permita comparar los resultados del análisis teórico y el práctico. Dicha comparación debe ser realizada y comentada, para entrega junto con el resto de requerimientos.

Para prevenir que más de un grupo o persona trabaje sobre el mismo problema, se proporciona una [hoja de cálculo electrónica en Google Sheets](#). Cada grupo o persona debe llenar los datos requeridos para dejar registro del problema a trabajar, de modo que ninguna otra persona o grupo elija el mismo problema. En Google Sheets se registra el historial de cambios y las personas que los realizaron, de modo que, si es necesario, se usará esta data para dar prioridad a las elecciones en orden cronológico.

Entregables

1. Problema elegido.
 - a. Definición y enunciado del problema. Cualesquiera ejemplos y materiales visuales que clarifiquen el problema contribuirán a una mejor calificación sobre este entregable.
 - b. Fuentes.
2. Algoritmos de solución: los algoritmos pueden ser desarrollados por usted (y su grupo), o pueden ser investigados.
 - a. Algoritmo DaC y explicación que justifique la aplicabilidad de este acercamiento, con respaldo en la teoría vista en clase.
 - b. Programa que implemente el algoritmo DaC.
 - c. Algoritmo de programación dinámica y explicación que justifique la aplicabilidad de este acercamiento, con respaldo en la teoría vista en clase. Para este entregable se aceptan algoritmos de tipo *top-down* con memoización o *bottom-up*.
 - d. Programa que implemente el algoritmo de programación dinámica.
 - e. Fuentes.
3. Análisis teórico.
 - a. Procedimiento de análisis y tasa de crecimiento (expresada con notación asintótica) del algoritmo DaC. Debe plantearse el tiempo de ejecución como una relación de recurrencia, y aplicarse (con justificación) cualesquiera de las técnicas de resolución de recurrencias vistas en clase (substitución, árbol de recursión o método maestro).
 - b. Procedimiento de análisis y tasa de crecimiento (expresada con notación asintótica) del algoritmo de programación dinámica. Cualquier técnica aplicable de análisis vista en clase es permitida.
4. Análisis empírico.
 - a. Listado de entradas de prueba usadas para medir tiempos de ejecución de cada algoritmo (deben usarse las mismas entradas para probar ambos algoritmos). Deben proporcionarse suficientes entradas para que la curva que se dibuje en la gráfica evidencie la tendencia.
 - b. Gráficas que presenten visualmente los tiempos de ejecución de cada algoritmo en función de las entradas de prueba.
 - c. Comentarios de comparación entre los tiempos de ejecución obtenidos con las entradas de prueba y la tasa de crecimiento alcanzada con el análisis teórico, por cada algoritmo. El propósito es respaldar los análisis teóricos con la evidencia empírica, y explicar brevemente las razones para cualquier discrepancia o diferencia que se aprecie.