Universidad del Valle de Guatemala
Análisis y Diseño de Algoritmos
Docentes: Paulo Mejía; Tomás Gálvez P.
Semestre 1, 2023

Proyecto #2

Introducción

El objetivo de este proyecto es repasar en la práctica los beneficios del acercamiento *greedy* para solución de problemas.

Las condiciones son las siguientes:

- Fecha de entrega: 02 junio 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 deberán investigar un problema que se pueda resolver con programación dinámica y cuya solución con programación dinámica tenga un tiempo de ejecución superpolinomial. Luego deberán investigar o desarrollar un algoritmo que busque resolver el problema con un acercamiento *greedy*.

Finalmente evaluarán el desempeño de ambos algoritmos de manera teórica y empírica, y los compararán en función de dos aspectos:

- Tiempo de ejecución para proveer una solución.
- "Calidad" de la solución.

El término de calidad de la solución hace referencia a qué tan cerca se encuentra la solución provista por *greedy* de la solución óptima real. Es complicado medir esta cercanía cuando computar la solución óptima real es, en sí, complicado o imposible. Por tanto, deberán sustentar su argumento proveyendo los siguientes análisis:

- Demostración de que el problema presenta subestructura óptima.
- Relación de recurrencia que describa la computación de la solución óptima.
- Discusión sobre la presencia de la *greedy choice property* en el problema (y demostración de su presencia, si se presenta).

Tome en cuenta que si su problema presenta la *greedy choice property* será posible demostrar que la presenta (e incluso desarrollar una solución) modelándolo como una matroide.

Universidad del Valle de Guatemala
Análisis y Diseño de Algoritmos
Docentes: Paulo Mejía; Tomás Gálvez P.
Semestre 1, 2023

Entregables

A continuación, el listado de entregables para el proyecto. El orden en el que se listan es una sugerencia de cómo trabajarlos para facilitar las discusiones.

1. **(5 pts.)** Descripción del problema elegido.
2. **(20 pts.)** Análisis del problema:
    a. **(10 pts.)** Demostración de que el problema presenta subestructura óptima.
    b. **(10 pts.)** Relación de recurrencia que describa la computación de la solución óptima.
3. **(40 pts.)** Algoritmos de solución:
    a. **(5 pts.)** Algoritmo de solución con programación dinámica en pseudocódigo.
    b. **(5 pts.)** Algoritmo de solución con acercamiento *greedy* en pseudocódigo.
    c. **(15 pts.)** Análisis de los algoritmos (*i.e.*, tiempo de ejecución de cada uno; investigado o calculado, pero explicado en ambos casos).
    d. **(15 pts.)** Implementación funcional de ambos algoritmos en el lenguaje de su elección.
4. **(10 pts.)** Discusión sobre presencia de *greedy choice property* (y demostración de su presencia, si se presenta).
5. **(15 pts.)** Análisis empírico:
    a. **(5 pts.)** Listado de entradas de prueba usadas para medir tiempos de ejecución de ambas implementaciones.
    b. **(5 pts.)** Diagrama de dispersión que muestre los tiempos de ejecución los programas en función de su tamaño de entrada.
    c. **(5 pts.)** Regresión polinomial que se ajuste mejor a los datos.
6. **(10 pts.)** Discusión final sobre la aplicabilidad del acercamiento *greedy* para resolver el problema elegido de manera óptima.

La entrega debe consistir en los archivos de código desarrollados (y, además, si se desea, enlace al repositorio donde se encuentra el código) y un documento PDF con los análisis y discusiones requeridas.