

TRABAJO PRÁCTICO 2

PROBLEMA 1 - Programación Dinámica

Cualquier cadena puede ser descompuesta en secuencias de palíndromos. Por ejemplo, la cadena ARACALACANA se puede descomponer de las siguientes formas:

ARA CALAC ANA
ARA C ALA C ANA
A R A CALAC A N A
etc.

Desarrollar un algoritmo de programación dinámica que encuentre el menor número de palíndromos que forman una cadena dada. Por ejemplo, para ARACALACANA debería devolver 3.

Se pide:

1. Supuestos: identificar supuestos, condiciones, limitaciones y/o premisas bajo los cuales funcionará el algoritmo desarrollado
2. Diseño:
 - a. Definir la Ecuación de Recurrencia
 - b. Justificar el cumplimiento de los requisitos de Subestructura Óptima y Subproblemas Superpuestos
 - c. Indicar cómo usa Memoization
 - d. Incluir un Pseudocódigo
 - e. Detallar las estructuras de datos utilizadas
3. Seguimiento: Ejemplo de seguimiento con un set de datos reducido
4. Complejidad: Análisis de la complejidad temporal a partir del pseudocódigo
5. Sets de datos: diseñar sets de datos apropiados. Se pueden generar utilizando una función random con una semilla fija, para permitir la reproducibilidad de los resultados, o ser generados manualmente e incluidos como archivos que lee el programa.
 - a. Cada set de datos debe ser incluido en la entrega, junto con el resultado obtenido en cada caso.
 - b. El programa entregado debe generar los sets de datos o leerlos desde archivos.
6. Tiempos de Ejecución: medir los tiempos de ejecución de cada set de datos y presentarlos en un gráfico.
7. Informe de Resultados: redactar un informe de resultados comparando los tiempos de ejecución. ¿Se corresponde con la complejidad determinada inicialmente?
 - a. El gráfico comparativo de tiempos debe incluir tanto la curva con los valores medidos como la curva teórica de ajuste.

PROBLEMA 2 - Programación Lineal

Concesiones Argentina 2000 SRL tiene la concesión de los espacios publicitarios en las paradas de colectivos de un municipio. Son en total 200 paradas y tiene ofertas de distintos productos para el próximo mes:

- Cliente A ofrece USD 50000 por ocupar 30 paradas
- Cliente B ofrece USD 100000 por ocupar 80 paradas o USD 120000 por 120 paradas (sólo una de ambas opciones)
- Cliente C ofrece USD 100000 por ocupar 75 paradas
- Cliente D ofrece USD 80000 por ocupar 50 paradas
- Cliente E ofrece USD 5000 por ocupar 2 paradas
- Cliente F ofrece USD 40000 por ocupar 20 paradas
- Cliente G ofrece USD 90000 por ocupar 100 paradas

Por ser competidores directos, no se puede hacer publicidad simultáneamente de los clientes A y D. Se desea determinar a qué clientes se concesionará la publicidad de las paradas para maximizar el beneficio total.

Se pide:

1. Supuestos. Indicar supuestos, condiciones o premisas bajo las cuales se plantea el modelo.
2. Variables. Definir qué mide cada variable, e indicar si es continua, entera o binaria/indicativa
3. Modelo de Programación Lineal. Definir y describir la función objetivo y las restricciones que tendrá el modelo. Todas deben ser lineales.
4. Solución:
 - a. Desarrollar un programa que resuelva el modelo usando Python y Pulp o un software de programación lineal (Lindo, GLPK, CPLEX, etc.).
 - b. Incluir todos los archivos necesarios para la ejecución
 - c. Incluir un archivo con el resultado obtenido
5. Informe:
 - a. Describir la solución hallada en términos del dominio del problema

PROBLEMA 3 - Redes de Flujo

Estamos construyendo una red WAN con n antenas y queremos que tenga un buen nivel de tolerancia a fallas. Dada una antena, su conjunto de backup de tamaño k es el conjunto de k antenas que se encuentran a una distancia menor a D . Queremos evitar que una antena pertenezca al conjunto de backup de más de b antenas, precisamente para evitar que un fallo pueda afectar a una porción importante de la red. Suponer que conocemos los valores D , b y k , y que tenemos una matriz $d[1..n, 1..n]$ con las distancias entre antenas, de forma tal que $d[i,j]$ es la distancia entre la antena i y la j .

Plantear un algoritmo de complejidad polinomial que encuentre el conjunto de backup de tamaño k de cada una de las n antenas, de forma tal que ninguna aparezca en más de b conjuntos de backup, o bien, que indique que no existe una solución posible. Debe apoyarse en el algoritmo estándar de Ford-Fulkerson, la variante escalada o la de Edmonds-Karp.

Se pide:

1. Supuestos: identificar supuestos, condiciones, limitaciones y/o premisas bajo los cuales funcionará el algoritmo desarrollado
2. Diseño:
 - a. Explicación en prosa de cómo se adaptan los datos de entrada a una red de flujo, y de cómo se debe interpretar la salida del algoritmo de Ford-Fulkerson para resolver este problema. Se recomienda incluir diagramas de apoyo.
 - b. Pseudocódigo y estructuras de datos utilizadas
3. Seguimiento: Ejemplo de seguimiento con un set de datos reducido.
4. Complejidad: Análisis de la complejidad temporal a partir del pseudocódigo
5. Sets de datos: diseñar sets de datos apropiados. Se pueden generar utilizando una función random con una semilla fija, para permitir la reproducibilidad de los resultados, o ser generados manualmente e incluidos como archivos que lee el programa.
 - a. Cada set de datos debe ser incluido en la entrega, junto con el resultado obtenido en cada caso.
 - b. El programa entregado debe generar los sets de datos o leerlos desde archivos.
6. Tiempos de Ejecución: medir los tiempos de ejecución de cada set de datos y presentarlos en un gráfico.
7. Informe de Resultados: redactar un informe de resultados comparando los tiempos de ejecución. ¿Se corresponde con la complejidad determinada inicialmente?
 - a. El gráfico comparativo de tiempos debe incluir tanto la curva con los valores medidos como la curva teórica de ajuste.

Condiciones Generales de Entrega

- El trabajo debe ser entregado en un archivo zip conteniendo:
 - Documento con carátula, índice y numeración de páginas. La carátula debe incluir nombre y padrón de los integrantes del grupo. Debe presentarse en formato PDF.
 - Archivos con los fuentes de los algoritmos desarrollados.
 - Archivo README indicando el lenguaje de programación utilizado, versión mínima y bibliotecas requeridas, e instrucciones para ejecutar.
 - Archivos con los sets de datos utilizados
 - Archivos con resultados obtenidos para cada set de datos
- IMPORTANTE: el tamaño de los sets de datos (cuando corresponda) debe ser tal que permita obtener suficiente información como para graficar los tiempos de ejecución y verificar la complejidad temporal.
- Se calificará positivamente la inclusión de referencias bibliográficas relevantes. Para ello, se pueden incluir citas en el texto del informe siguiendo el modelo propuesto por Rivas (2022) y luego incorporar el listado completo en un anexo al final, usando normas APA 7ma edición.

Referencias:

- Rivas, A. (2022, mayo 25). Cómo hacer una lista de referencias con Normas APA. Guía Normas APA. <https://normasapa.in/como-hacer-la-lista-de-referencias/>