UNAHUR
UNIVERSIDAD NACIONAL
DE HURLINGHAM

Trabajo Práctico Algoritmos

Sistema de Gestión de Envíos y Logística

Descripción General

El sistema debe permitir gestionar una red de centros de distribución y envíos en distintas ciudades. Se deberá implementar en C++ de forma completamente orientada a objetos, utilizando estructuras eficientes y haciendo uso adecuado de la memoria dinámica.

Archivos de entrada

- **centros.txt**

  Cada línea contiene:

  código nombre ciudad capacidad paquetes_diarios empleados

Ejemplo:

CBA CentroCordoba Córdoba 5000 2000 120

MZA CentroMendoza Mendoza 3000 1200 80

- **envios.txt**

  Cada línea representa un envío procesado:

  codigo_centro paquete_id cliente_id fecha_dia peso

Ejemplo:

CBA 1245 302 20250412 3.5

MZA 1250 411 20250215 2.0

- **conexiones.txt**

  Los pesos son ser la distancia entre ciudades.

  Crear un archivo con: *centro_origen centro_destino distancia.*

Ejemplo:

CBA MZA 900

UNAHUR
UNIVERSIDAD NACIONAL DE HURLINGHAM

CBA BUE 700

MZA BUE 1100

Funcionalidades requeridas

A. Gestión de Centros

- Mostrar la información de un centro de distribución específico.
- Agregar un nuevo centro.
- Eliminar un centro existente.
- Mostrar todos los centros (ordenados por capacidad, paquetes procesados o cantidad de empleados).
- Dado un centro inicial y uno destino, calcular el camino mínimo para transferir un paquete.

Requisito: Utilizar una tabla de hashing con una función hash propia. Justificar la elección de la función hash.

B. Análisis de Envíos

1. Calcular el total de envíos por centro en un rango de fechas dado.
2. Detectar centros con sobrecarga (más de X envíos en una semana).
3. Buscar todos los envíos de un cliente por identificador de paquete.

Consigna de análisis: Para cada algoritmo implementado en esta sección, indicar su complejidad temporal en notación Big O y justificar si puede mejorarse.

C. Optimización con Backtracking

Cada camión de reparto tiene una capacidad máxima de carga (en kg).
Se debe decidir qué paquetes cargar en un camión para maximizar el valor de entrega (prioridad o beneficio) sin superar la capacidad.

- **Entrada**: lista de paquetes con peso y valor.

- **Salida**: subconjunto de paquetes que maximiza el valor total.
- Implementar una solución con Backtracking.
- Indicar la complejidad temporal y discutir posibles optimizaciones.

Condiciones de entrega

- Exposición semanal de avances.
- Entrega final con defensa oral del proyecto.
- Presentación del diseño UML.
- Revisión del uso de estructuras de datos y justificación algorítmica.
- Debe implementar las clases de manera totalmente original y propia. La detección de copias resultará en la **pérdida de la regularidad de los involucrados**.
- El código debe estar escrito en **camelCase**. Los nombres de las clases se escriben en **PascalCase** (todas las palabras con mayúscula).