

Licenciatura en Sistemas

Tecnicatura en Informática

Trabajo Práctico nº3

Algoritmo Goloso: El Censo

Programación III

1º semestre - 2022

Docente: Daniel Bertaccini

Integrantes: Alan Fernandez
Soledad Ramirez

Com: 02

Introducción:

El trabajo práctico consiste en implementar un algoritmo goloso para asignar a un conjunto de censistas “manzanas a censar” de un denominado “radio censal”, el cual está representado por un grafo donde los vértices simbolizan las manzanas, y las aristas relacionan las manzanas que se encuentran contiguas en el mapa. Se cuenta con la condición de que las manzanas asignadas a un censista deben ser vecinas, con un máximo de tres manzanas por censista y que se debe minimizar el número de censistas que visitarán las manzanas del radio censal.

Desarrollo del ejercicio:

Para desarrollar el ejercicio decidimos asignar los censistas de izquierda a derecha en la primera fila, para luego recorrer la segunda fila de derecha a izquierda y volver a cambiar de sentido en la tercera fila y así sucesivamente. Esto nos permite asignar las manzanas minimizando la cantidad de censistas.

Para organizar el proyecto realizamos el desarrollo del mismo utilizando el patrón MVC el cual separa las capas de vista, interacción con del programa con el usuario y la lógica de negocios.

Paquete Vista:

En el **main** del proyecto se inicializa una ventana de **inicio** desde la cual se inicializa el ejercicio. Se direcciona a otra vista de la clase **Marco**, dentro de la cual podemos configurar los datos del programa.

Por un lado contamos con el botón **Agregar_censista** el cual redirige a un formulario de la clase **FormCensista** donde es posible ingresar el nombre, el apellido y la edad del censista. Además, dentro de este formulario implementamos un **JFileChooser** el cual nos permite agregar una imagen a cada censista a través de la clase **CargaFoto**.

Asimismo, dentro del **marco** se añadió el botón **Configurar_radio_censal** el cual permite ingresar (a través de **JOptionPane**) dos enteros que representan la cantidad de manzanas que conformarán el radio censal de manera horizontal y vertical.

Una vez añadidos los censistas y configurado el radio censal, se puede optar por los botones **Algoritmo_goloso** o **Fuerza_bruta** para resolver el ejercicio con cada uno de esos procedimientos. El resultado se muestra dentro de un **JScrollPane** dentro del **marco**.

Por último, se puede ver un detalle de los censistas con el botón **Ver_detalle** donde se muestran los datos completos de cada uno de los censistas y su imagen con la vista **DetalleCensista**.

Además, con el botón **Ver_mapa** podemos observar el radio censal en un mapa real.

Paquete Modelo:

Para comenzar desarrollamos la clase **Grafo** que nos permite representar las manzanas y sus aristas. En el constructor se inicializa el **ArrayList vértices** y cada **HashSet** que compone dicha lista donde se configurarán los vecinos de cada manzana.

Además contamos con métodos para trabajar con el grafo:

- **agregarArista()**: agregamos los pares de vecinos en sus respectivos **HashSets**
- **removeArista()**: remueve los dos pares de vecinos del array **vértices**.
- **existeArista()**: método booleano que indica si existe la arista.
- **tamaño()**: devuelve la cantidad de vértices (tamaño) del array de vértices del grafo.
- **crearGrafoCenso()**: este método setea las aristas del grafo creado, agregando aristas horizontales en las manzanas que se encuentran en la misma fila y también agrega las aristas verticales en manzanas que se encuentran en la misma columna para formar el radio censal.

También contamos con la clase **Censista** en el cual se cuenta con datos del censista como su id, nombre, apellido y edad además de una imagen relacionada. También presenta un **ArrayList misManzanas** en donde se guardarán las manzanas asignadas para censar con el método **cargarManzanas()**. Cuenta con un método **mostrarInfo()** el cual se utiliza para mostrar la solución en el marco. También el método **mostrarManzanas()** se utiliza para mostrar el detalle de cada censista. Por último el getter de la imagen para poder mostrarla por pantalla.

Implementamos una clase **Negocio** la cual contiene la variable estática **misCensistas** la cual cuenta con un **ArrayList** de todos los censistas ingresados por el usuario, el cual es utilizado por diferentes clases además cuenta con getters y setters del ancho y alto del radio censal, los cuales son utilizados para configurar el radio censal.

La clase **Mapa** implementa un **JmapViewer** el cual muestra por pantalla un mapa donde se inserta el radio censal configurado por el usuario.

También encontramos la clase **AlgoritmoGoloso** donde se asignan las manzanas (con el método **asignarManzanas()**) a los censistas recorriendo el radio censal de izquierda a derecha en la primera fila y al bajar a la siguiente fila se cambia el sentido del recorrido (de derecha a izquierda), siguiendo ese criterio en todas las filas siguientes. Con esto logramos minimizar la cantidad de censistas asignados. Este método utiliza el método privado sobrecargado **asignarManzanas(Censista, int)** el cual usa el método de **Censista cargarManzanas()** para asignar la manzana al censista considerado. También el método **SwitchIzqDer()** el cual según el signo del int **bajéEnElGráfico** recorrerá esa porción del radio censal hacia la derecha o hacia la izquierda según corresponda.

Construimos además la clase **AlgoritmoFuerzaBruta** donde contamos con otro procedimiento para asignar las manzanas a censar a los censistas. En este caso se generan todas las combinaciones posibles con los vértices del grafo con un método recursivo llamado

generarSubconjuntos() y se los guarda en HashSet los cuales se agrupan dentro de una lista denominada **conjuntosManzanas**. El método **asignarManzanas()** implementa el método sobrecargado **asignarManzanas(Censista)** el cual asigna las manzanas a los censistas siempre que la manzana no haya sido asignada anteriormente, que el censista no tenga asignada las tres manzanas y que la manzana a asignar sea vecina de las manzanas asignadas al censista hasta ese momento. Para esto se utiliza el método **manzanasDisponibles()**, **sonVecinas()**, **estanAsignadas()** y **estaAsignada()**.

Paquete Controller:

Dentro de este paquete desarrollamos las clases que manejan los eventos realizados por el usuario dentro del programa.

La clase **ControladorAgregarCensistas** controla el comportamiento del botón **Agregar_censistas** del marco. El cual lanza un nuevo **formCensista** donde se accede a un formulario que permite ingresar los datos del censista y su foto. El botón volver permite cerrar la ventana y volver a la vista **marco**.

Dentro del formulario **formCensista** se controla el botón **Agregar_datos** a través de la clase **ControladorDatosCensista**, el cual captura los datos ingresados por el usuario en los inputs, genera una nueva instancia de **Censista** y la agrega a la lista estática **censistas** de la clase **Negocio**, por último se resetean los inputs para un nuevo ingreso.

La clase **ControladorCrearRadio** maneja las acciones que se disparan al presionar el botón **Configurar_radio_censal**. Lanza **JOptionPanes** para que el usuario ingrese el número de manzanas que tendrán las filas del radio censal y luego el número de columnas. Con estos datos el grafo generará las aristas horizontales y verticales con el método **crearGrafoCenso()** y así construir el radio censal.

El proyecto también cuenta con la clase **ControladorAsignarManzanas** el cual implementa el algoritmo goloso construido y lo muestra en pantalla dentro de un **ScrollPane** dentro del **marco**.

Por otro lado el botón **Fuerza_bruta** es manejado por la clase **ControladorAlgoritmoFuerzaBruta**, el cual implementa el **AlgoritmoFuerzaBruta** y muestra la solución por pantalla dentro del **marco**.

A partir de la solución se puede ver el radio censal construido dentro de un mapa real utilizando el botón **Ver_mapa** a través de la clase **ControladorMapa**.

Finalmente, una vez obtenida la solución podemos acceder al detalle de los censistas, sus datos, su foto de perfil y las manzanas que se le asignaron usando el botón **Ver_detalle**, el cual abre la vista **DetalleCensista**.

Dentro de la vista **DetalleCensista** se genera una instancia de la clase **ControladorSiguienteCensista** el cual controla el botón siguiente, que permite visualizar

todos los censistas de la lista estática **Censistas** y sus datos, así como también su foto de perfil.

Paquete Test:

Dentro de este paquete incluimos dos clases que corresponden a los tests de la clase Grafo, el procedimiento heurístico implementado y el procedimiento de fuerza bruta.

Conclusiones:

Con este trabajo implementamos dos diferentes procedimientos para resolver el problema de la asignación de las manzanas a los censistas a partir de un radio censal representado por un grafo. Tanto la heurística como el algoritmo de fuerza bruta nos permitieron encontrar un resultado óptimo en este ejemplo puntual, pero con diferente complejidad, por lo que concluimos que el algoritmo goloso es más eficiente que la aplicación de fuerza bruta. Pudimos aplicar estos algoritmos en un problema real que se puede resolver utilizando grafos y pudimos notar la importancia de su aplicación en la resolución.

Diagrama de clases:

