

Enunciado del problema (Space travellers)

Desde el inicio de la industrialización, la humanidad se ha visto involucrada en cientos de problemas: desde pobreza, hambrunas y sobre población, hasta pandemias que han afectado a millones de familias alrededor del mundo. También se han visto extinciones completas de algunas razas zootécnicas, e inclusive inundaciones de ciudades enteras debido al derretimiento de los polos. Además, los malos gobiernos y las dictaduras han provocado migraciones masivas a los sitios con mayor posibilidad de sobrevivir. Es decir, los países potencia.

Esto ha llegado a tal punto que en el año 2157 una de las soluciones más viables para salvar la raza humana es embarcarse en una misión para explorar los confines de la Vía Láctea. Es por esta razón, que las principales naciones en este planeta se han unido y han conformado una sola nación llamada “La Tierra Unida”. Bajo esta nueva unión se plantea la creación de una nave interplanetaria llamada USS DISCOVERY, con la idea de recorrer el espacio para visitar los posibles lugares habitables y recibir el apoyo de otras civilizaciones.

Al ser un proyecto tan ambicioso e importante, se necesita de las mentes más creativas, innovadoras y progresistas para llevarlo a cabo. Por ende, han decidido solicitar ayuda a 3 jóvenes apasionados por las estructuras de datos para desarrollar el software de la nave. Cuando llega el tiempo de la primera reunión, sus intelectos lo llevan a imaginar un sistema completo de navegación en la nave, basado a partir de una base de datos de sistemas planetarios existentes, claramente con rutas de navegación entre ellos.

Gracias a información recolectada con anterioridad, los 3 jóvenes saben que cada sistema planetario cuenta como mínimo con una estrella y un planeta, de los cuales se almacenan únicamente los nombres. Además, con apoyo de otras civilizaciones avanzadas que se han comunicado con la Tierra, se sabe que una cantidad considerable de sistemas ya se encuentran habitados por alguna civilización. De los sistemas se desea almacenar el nombre, nombres de civilizaciones que lo habitan (si es que hay), fecha de descubrimiento y coordenadas.

Ahora bien, como se posee una gran cantidad de datos e información, uno de los integrantes plantea indicarle a la nave el recorrido con menor tiempo para viajar del sistema actual a otro. Todos lo miraron seriamente por tan magnífica solución, y entendieron que el trabajo grupal es mágico.

Ahora bien, como este es un grupo competitivo y todos quieren proponer ideas o soluciones eficaces, cada uno se toma la libertad de implementar algo nuevo: Primero, una pantalla de búsqueda que le permita seleccionar un sistema al ingresar su nombre.

Segundo, si la tierra unida descubre un nuevo sistema, contactará a la tripulación de la nave para que esta lo agregue a la base de datos, ya que el sistema de comunicación de la nave no permite agregarlo de forma automática. Por último, si algún evento cósmico destruye un sistema este debe poder ser eliminado del software de navegación de la nave.

Pero, como es un programa de gran importancia no sólo para la nave, sino para la humanidad también, estas brillantes mentes deciden probar las capacidades del programa durante el viaje, generando aleatoriamente una base de datos de prueba con al menos 100.000 sistemas planetarios que luego puede ser cargada en el programa.

Requerimientos

Requerimientos Funcionales

1. Calcular el recorrido que toma menos tiempo para viajar del sistema actual al sistema deseado.
2. Agregar un sistema planetario nuevo al programa y a su base de datos.
3. Eliminar un sistema planetario del software.
4. Implementar una pestaña donde se pueda buscar un sistema planetario para ver su información.
5. (Opcional) Que cada campo de búsqueda despliegue una lista de autocompletar.

No funcionales

1. Modelar e implementar el sistema de navegación en dos grafos diferentes.
2. Implementar la interfaz gráfica con JavaFx.
3. Contar con un menú superior en la ventana, el cual permite cambiar todos los elementos cuando se esté trabajando en opciones diferentes.
4. Implementar los algoritmos de recorridos sobre grafos BFS y DFS.
5. Codificar los algoritmos de camino de peso mínimo Dijkstra y Floyd-Warshall.
6. Desarrollar los algoritmos de árbol de recubrimiento mínimo Prim y Kruskal.
7. Generar sistemas de manera aleatoria.