**Enunciado**

El problema a resolver se encuentra descrito en el juez en línea de la universidad de Valladolid (Uva), lo puedes encontrar en el siguiente [enlace](https://uva.onlinejudge.org/external/113/11367.pdf)

**Qué debe llevar a cabo como parte del proyecto de curso?**

1. Desarrollar el TAD Grafo:
   1. Definir formalmente el TAD Grafo.
   2. Diseñar e implementar el TAD Grafo como una lista de **adyacencia**.
   3. Diseñar e implementar el TAD Grafo como una matriz de adyacencia.

El diseño de cada grafo debe incluir: el diagrama de clases del modelamiento del grafo, **el diseño de casos de pruebas unitarias automáticas y el diagrama de clases de las pruebas unitarias automáticas**.

1. Con base en su implementación del TAD Grafo resuelva el problema en UVa y obtenga Accepted. Usted debe proponer dos soluciones:
   1. Una utilizando su implementación del grafo que utiliza listas de aristas, y
   2. Otra con la implementación que utiliza matriz de adyacencia.

En ambos casos debe obtener **Accepted** como veredicto.

El enlace con el enunciado y la opción para enviar la solución es el siguiente: [enlace](https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=2352)

1. Implemente un generador de casos de prueba para el problema planteado. El caso de prueba generado aleatoriamente cada vez debe ser almacenado en un archivo de texto plano con las mismas especificaciones de la entrada indicadas en la descripción del problema. El programa que genera los casos de prueba debe ofrecer una interfaz gráfica que permita, una vez generado un caso de prueba, indicar donde será almacenado el archivo y con qué nombre.
2. Debe desarrollar un programa con interfaz gráfica (es decir, que permita una adecuada interacción con un usuario estándar) y un modelamiento de mundo e interfaz de acuerdo con las buenas prácticas de programación. Este programa debe tener las siguientes funcionalidades:
   1. Permitir cargar el archivo de entrada a través de un archivo de texto. El usuario indicará a través de la interfaz gráfica la ubicación y nombre del archivo de texto a cargar. El formato de la entrada será el mismo definido en la descripción del problema.
   2. Una vez cargado el archivo, desplegar un componente gráfico a través del cual se pueda elegir uno de los casos de prueba del archivo cargado, y una vez se elija el caso de prueba visualizar un mapa que indique claramente el precio del combustible en las ciudades, los caminos.
   3. Al visualizar el caso de prueba, desplegar (o activar) un botón que entregue en un campo de texto (como parte de la misma ventana del programa) la solución de ese caso de prueba en el mismo formato especificado en la descripción del problema.
   4. Permitir visualizar igualmente la salida o solución del problema en la interfaz gráfica. Esto puede hacerse, por ejemplo, resaltando el camino que toma el carro y especificando en cuales ciudades tanqueo.
   5. Agregar un botón que entregue un informe detallado para cada salida o solución del problema. En donde se muestre en qué ciudades agregó combustible al carro, cuantos galones y la cantidad de dinero pagado en la estación del pueblo.
3. (Bonus) Se entregarán puntos extra en la calificación del proyecto si su solución es capaz de resolver casos de prueba con n =10^6 y m=10^7. Para que el bonus sea válido se deben generar varios casos de prueba que cumplan con estas especificaciones y que su algoritmo entregue una solución correcta en un tiempo prudente.

**Condiciones**

* El proyecto se puede desarrollar en grupo de máximo 2 personas.
* La entrega completa y final del proyecto deberá hacerse a más tardar el Lunes 22 de Mayo de 2017 a las 23:55 a través de moodle y una sustentación al profesor cuyo horario será programado con cada grupo.
* Habrá una entrega parcial el viernes de cada semana que sigue a la publicación de este enunciado y el avance es el punto completamente resuelto de la semana equivalente. Es decir, la primera semana se deberá entregar el punto 1 completo, la segunda semana el punto 2 completo y así hasta el final.

Que lo disfruten!