

2019 1212-Final V2 - examen

Programación avanzada (1113)

Programación Avanzada (1113)
Fecha: 16/12/2021

UNLaM
Instancia: Final

Consigna general

- Responder todas las preguntas en un documento .pdf.
- El nombre de los archivos debe ser: DNI-ApellidosNombres.
- La entrega se realizará a través de la solapa "Prácticas" de la plataforma MleL.

1. Java

Se va a implementar un simulador de Vehículos. Existen dos tipos de Vehículo: Coche y Camión.

1. Sus características comunes son la matrícula y la velocidad. En el momento de crearlos, la matrícula se recibe por parámetro y la velocidad se inicializa a 0. El método toString() de los vehículos devuelve información acerca de la matrícula y la velocidad. Además se pueden acelerar, pasando por parámetro la cantidad en km/h que se tiene que acelerar.
2. Los coches tienen además un atributo para el número de puertas, que se recibe también por parámetro en el momento de crearlo. Tiene además un método que devuelve el número de puertas.
3. Los camiones tienen un atributo de tipo Remolque que inicializa a null (para indicar que no tiene un remolque). Además, tiene un método ponRemolque(), que recibe el Remolque por parámetro, y otro quitaRemolque(). Cuando se muestre la información de un camión que lleve remolque, además de la matrícula y velocidad del camión, debe aparecer la información del remolque.
4. En esta clase, el método acelerar además debe lanzar una excepción de tipo DemasiadoRapidoException. si el camión tiene remolque y la velocidad más la aceleración superan los 100 km/h
5. La clase Remolque tiene un atributo de tipo entero que es el peso y cuyo valor se le da en el momento de crear el objeto. Debe tener un método toString() que devuelva la información del remolque.

Se pide:

- a) Hacer el diagrama UML del simulador de vehículos
- b) Implemente el método acelerar en todas las clases que corresponda
- c) Implementar la clase DemasiadoRapidoException.

2. Prolog

Tomas, Linda, Susana, Javier y Daniel van a una cena.

Tomas conoce a Susana y Linda.

Daniel conoce a Susana y Linda.

Javier conoce a Daniel y Linda.

¿Es posible sentarlos en una mesa redonda de manera tal que cada uno tenga a sus lados a personas que conozca?

En caso de ser posible, escriba las **reglas prolog** que permiten listar todas las maneras en que estos amigos podrían sentarse a la mesa.

3. Grafos

Dado el problema OIA_ Iluminación de un Parque, cuyo enunciado se encuentra a continuación, Responder las siguientes preguntas:

Prueba

1. ¿Qué método(s) de prueba utilizaría para probar la solución de este problema? Puede seleccionar más de una opción. Justificar.
 - Programa probador
 - Lote de prueba
 - Pruebas unitarias
 - Lote de prueba de programa probador

Implementación

2. ¿Cuál de los siguientes algoritmos utilizaría para resolver este problema? Justificar.
 - Dijkstra
 - Kruskal
 - Búsqueda en matrices
 - BFS
 - Coloreo
3. ¿Cuál de las siguientes estructuras de datos utilizaría para resolver/optimizar la solución? Puede seleccionar más de una opción. Justificar
 - Array bidimensional (matriz)
 - Lista
 - Cola de prioridad
 - Programación dinámica
 - Union-Find
 - Pila
4. ¿Cómo calcularía el costo de cada una de las aristas?
5. ¿Qué nivel de conectividad tendría el grafo resultante en memoria antes de aplicar el algoritmo para la solución? Justificar.

Complejidad Computacional

6. ¿Qué complejidad computacional tendría la solución propuesta? (debe corresponderse con las opciones seleccionadas anteriormente). Justificar.
 - $O(N^2)$
 - $O(A \cdot \log(N))$
 - $O(N)$
 - $O(N+A)$
 - $O(N \cdot \log(N))$
 - $O(N \cdot \log(A))$
 - $O(N^3)$
 - Otra-¿Cuál?

Iluminación de un Parque

Contribución de Jorge Arroyuelo

Descripción del problema

En un importante parque de la ciudad se esta realizando una obra de iluminación, por lo que se han instalado una serie de faroles distribuidos sobre la superficie del mismo, numerados y localizados mediante coordenadas (a,b) .

El problema con esto es que, debido al tamaño del parque y a la cantidad de faroles necesarios para su iluminación, esta obra resulta demasiado costosa de realizar. De esta forma, un análisis de la situación nos lleva a concluir que la única manera posible de reducir costos es economizar la cantidad de cable que deberá ser empleado para interconectar los faroles.

Se debe tener en cuenta que la red eléctrica que conectará todos los faroles no puede tener empalmes en cualquier lugar, sino que estos deben realizarse en las bocas correspondientes situadas en cada farol. De esta manera, cada farol se debe conectar directamente a otro, donde se realiza el empalme necesario para obtener la alimentación eléctrica. Al menos uno de los faroles debe conectarse al tablero principal, el cual se encuentra ubicado en la entrada del parque e indica el punto inicial a partir de donde se toman las coordenadas, es decir que el tablero se encuentra ubicado en el punto con coordenadas $(0,0)$. Para realizar el trabajo los cables siempre son cortados en longitudes enteras, dejando de esta forma un "CHICOTE" en ambos extremos para facilitar el trabajo en las bocas. La suma de las longitudes de los chicotes da un largo L ($1 < L \leq 2$).

Se te pide que escribas un programa `parque.cpp`, `parque.c` o `parque.pas` que determinen como deben ser interconectados los faroles con el objetivo de utilizar la menor cantidad total de cable.

Datos de entrada

Se recibe un archivo `parque.in` con el siguiente formato:

- Una línea que indica la cantidad N ($1 \leq N \leq 1.000$) de faroles a instalar en el parque.

- N líneas con 3 números n, a, b ($1 \leq n \leq N, 1 \leq a, b \leq 100.000$) que indican el numero de farol y su posición (coordenadas) dentro del parque.

Datos de salida

Se debe generar un archivo `parque.out` conteniendo:

- Una línea indicando la cantidad de cable necesaria para realizar la obra completa.
- Una línea por cada farol indicando en primer lugar el numero del mismo y luego una secuencia de números correspondientes a los faroles (0 en caso de ser el tablero principal) con los cuales se encuentra conectado directamente, separados por espacio. Estas líneas estarán ordenadas entre sí por número creciente de farol.

En la siguiente página se muestra un ejemplo de solución de este problema

Puntuación

Se recibirá la mitad del puntaje por la primera línea (longitud total) y la otra mitad por el cuadro de conexiones.