



CONTROL 3 ESTRUCTURAS DE DATOS 2022-2

Fecha de Aplicación: 11 de noviembre del 2022

Objetivos de la evaluación:

El siguiente control tiene como objetivo medir los aprendizajes de los estudiantes en los siguientes aspectos:

- Resolver problemas computacionales utilizando pseudocódigo y lenguaje C utilizando tipos de datos abstractos (TDA), estructuras de datos lineales y no lineales (grafos).
- Aplicación de estructuras de datos listas, colas, pilas y grafos en sus distintas versiones de representatividad.

Instrucciones:

1. El control estará habilitado 24 horas a partir del viernes 11 de noviembre a las 16:00 horas, por lo tanto, podrá subir su respuesta a más tardar a las 16 horas del sábado 12 de noviembre.
2. Para resolver el problema del control debe considerar lo siguiente:
 - a. Antes de realizar el programa **haga un algoritmo en pseudocódigo** que resuelva el problema. Indicar claramente las entradas y salidas, siguiendo la estructura de algoritmo enseñada. Asuma que las operaciones elementales de los TDA que utilice están implementadas (sólo para esta parte). Recuerde de especificar las estructuras de datos utilizadas.
 - b. **Crear un programa en lenguaje c ANSI** para resolver el problema basado en el algoritmo presentado en a. Será probado en ambiente linux y windows.
 - c. Utilice las estructuras de datos y algoritmos que ha estudiado en esta asignatura junto con la bibliografía del programa curso.
 - d. Puede utilizar cualquiera de las estructuras de datos lineales y no lineales vistas en esta asignatura. Debe justificar y explicar muy bien las estructuras de datos que decida utilizar para resolver el problema.
 - e. El programa debe estar comentado, ordenado, fácil de leer y comprender.
3. Usted debe subir a la plataforma virtual un archivo **pdf** con el algoritmo en pseudocódigo, los archivos con extensiones **.c** y **.h** que implementan los TDA utilizados y el archivo **.c** con el programa fuente que resuelve el problema:
 - a. Formato nombre archivo pdf: C3_SECC_NombreApellido_RUT.pdf (ejemplo: C3_A1_JuanDiaz_20023066-3.pdf).
 - b. Formato nombre de los archivos .h: C3_SECC_TDAnombre.h (ejemplo: C3_A1_TDAgrago.h)
 - c. Formato nombre de los archivos .c que implementan los TDA utilizados: C3_SECC_TDAnombre.c (ejemplo: C3_A1_TDAgrago.c).
 - d. Formato nombre del programa fuente .c que resuelve el problema: C3_SECC_NombreApellido_RUT.c (ejemplo: C3_A1_JuanDiaz_20023066-3.c)
4. Recuerde que la copia (o plagio) no acredita que usted haya logrado los aprendizajes que se están evaluando en este control, por lo tanto, no se arriesgue a ser calificado con nota 1.0. Resuelva el problema por su propia cuenta e individualmente. No busque soluciones en internet u otras fuentes. El control es individual.



Problema:

El ministerio de salud habilitará en la región metropolitana, N centros de atención ambulatoria para cubrir necesidades específicas de atención al público. Una vez que estos centros estén operativos, el encargado de logística planificará sus visitas de control y supervisión a cada uno de los centros. El encargado pretende visitar los centros en días diferentes, por lo tanto, la salida la realizará siempre desde la municipalidad (origen). Se le solicita a usted crear el pseudocódigo y el programa C del algoritmo que mejor represente la resolución del problema antes descrito. Su implementación debe responder las siguientes preguntas:

- ¿Cuáles son las rutas mínimas y costos de traslado desde la municipalidad a cada uno de los centros de atención?
- ¿Cuál es el centro de atención con la ruta mínima que posee el mayor costo de traslado?

La información de entrada viene dada en un archivo con los datos de un grafo no dirigido.

Formato archivo de entrada:

n m //n número de nodos y m número de aristas
i j w //arista de i a j con peso w (valor real)

Formato Salida del programa:

Recorrido mínimo y costo:

origen, ..., n1: costo_camino_origen-n1

origen, ..., n2: costo_camino_origen-n2

...

Ruta mínima con el mayor costo de traslado:

origen, ..., nx: costo_camino_origen-nx

Ejecución del programa por línea de comando:

>programa nombre_archivo nodo_origen

Ejemplo:

Archivo de entrada: (ejemplo1.txt)

```
6 11
0 1 0.40
1 2 0.15
2 3 0.09
4 3 0.16
3 5 0.38
3 0 0.10
0 5 0.20
5 4 0.21
1 4 0.32
4 2 0.32
5 1 0.15
```



Salida del programa:

Recorrido mínimo y costo:

0, 3, 2, 1: 0.34

0, 3, 2: 0.19

0, 3: 0.10

0, 3, 4: 0.26

0, 5: 0.20

Ruta mínima con el mayor costo de traslado:

0, 3, 2, 1: 0.34

Ejecución:

>programa ejemplo1.txt 0

Criterios de evaluación:

Algoritmo (30 puntos):

Item I (10 pts): El algoritmo propuesto apunta a resolver el problema planteado, y **explica y utiliza** las estructuras de datos adecuadas y las restricciones solicitadas.

Item II (15 pts): Cumplido total o parcialmente el ítem I, el algoritmo resuelve correctamente el problema planteado.

Item III (5 pts): Cumplido total o parcialmente el ítem I, el algoritmo está escrito en pseudocódigo ordenado y consistente, identificando entradas y salidas

Programa (70 puntos):

Item I (10 pts): Consistente con el algoritmo entregado.

Item II (10 pts): Definición del TDA (**define y comenta** las estructuras y algoritmos vistos en clases que decidió utilizar para resolver el problema siendo consistente con lo solicitado)

Item III (40 pts): Cumpliendo total o parcial con la consistencia, TDA y estructuras de datos y algoritmos requeridos, la ejecución de su programa es correcta y usa las operaciones de la estructura de datos seleccionada (compila y entrega los resultados esperados)

Item IV (10 pts): Buenas prácticas de programación (ordenado, comentado, indentado, etc.)