Tecnológico de Monterrey, Campus Monterrey

Departamento de Ciencias Computacionales

**ANALISIS Y DISEÑO DE ALGORITMOS** - Ing. Román Martínez M.

**PRIMER EXAMEN DE PROGRAMACIÓN**  - 3 de Octubre de 2014

# Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Matrícula: \_\_\_\_\_\_\_\_\_\_\_\_

**Instrucciones:**

* Para resolver este examen puedes utilizar el compilador de C++ de tu preferencia para programar tus respuestas.
* Se te provee de un archivo llamado EX1.cpp en donde se encuentra cierto código que utilizarás y en ese mismo archivo programarás tus respuestas.
* Identifica tu archivo con un comentario inicial con tu nombre y matrícula, y documenta el código con los comentarios que consideres importantes y necesarios para explicar lo que programas.
* El archivo final de entrega como respuesta de tu examen, deberá subirse a la página del curso en el espacio correspondiente.
* Adicionalmente, deberás subir las respuestas a las preguntas de comprobación de funcionamiento que estarán en una liga aparte en la página del curso. Pide información al profesor si lo necesitas.
* Está estrictamente PROHIBIDO el acceso a códigos previamente realizados y cualquier tipo de comunicación electrónica o presencial. Cualquier situación deshonesta será penalizada con una calificación de DA.

Revisa el código fuente que se encuentra en el archivo EX1.cpp. Podrás observar que se encuentran definidos algunos arreglos que utilizarás y que ya está la implementación del algoritmo de Floyd. Usa este código para resolver los problemas II y III que se presentan a continuación. El problema I no requiere de ningún código adicional. Para probar el programa, deberás seguir las instrucciones que se indican y responder las preguntas asociadas a cada problema en la liga de captura del examen. El ALGORITMARIO contiene únicamente el algoritmo de la búsqueda binaria por si te fuera útil.

***PROBLEMA 1.*** *(40 puntos – Tiempo sugerido para solucionar el problema: 20 min).*

La **secuencia loca** es una serie de números en la que cada número de la serie, se calcula con la suma de los tres números anteriores. La siguiente definición recursiva, determina la forma de obtener el n-ésimo elemento de la secuencia loca:

**L(n) = 1 para valores de n = 1, 2 y 3.**

**L(n) = L(n-1) + L(n-2) + L(n-3)**

1. Basado en esta definición recursiva, aplica la técnica de la **programación dinámica** para diseñar el algoritmo de una función que sirva para obtener el n-ésimo elemento de la secuencia loca. Implementa este algoritmo y pruébalo.
2. Basado en esta definición recursiva, aplica la técnica de **divide y vencerás** para diseñar el algoritmo de una función que sirva para obtener el n-ésimo elemento de la secuencia loca. Implementa este algoritmo y pruébalo.

En la liga de captura del examen, deberás contestar las siguientes preguntas y anexar los códigos que programaste:

¿Cuál es el \_37\_\_\_-ésimo número de la secuencia loca? \_1467182629\_\_ ¿con qué algoritmo se obtiene más rápido? Con el iterativo (DP)

¿Cuál es el orden de complejidad del algoritmo implementado con programación dinámica? \_\_\_\_n\_\_\_\_\_

¿Cuál es el orden de complejidad del algoritmo implementado con divide y vencerás? \_\_\_exponencial\_\_\_\_\_\_

***PROBLEMA 2.*** *(30 puntos – Tiempo sugerido para solucionar el problema: 25 min).*

Utiliza la técnica de diseño de algoritmos más apropiada para obtener un algoritmo de orden logarítmico que resuelva el siguiente problema:

Un arreglo contiene la secuencia de números enteros que corresponde a los datos de las ventas que tuvo cierto negocio en los 365 días del año. Se sabe de antemano que el negocio tenía una tendencia a ir incrementando día a día sus ventas desde el primer día del año hasta que ocurrió un incidente que lo afectó y su ventas se cayeron, mostrando un decremento día con día hasta terminar el año.

El arreglo llamado ventas que tienes en el código del archivo EX1.cpp ya tiene estos datos. Implementa una función con tu algoritmo diseñado que sirva para encontrar el día del año en que ocurrió el incidente que provocó que las ventas se cayeran permanentemente (este día es el último en que hubo incremento en las ventas). Prueba tu implementación con el arreglo dado y responde lo siguiente en la liga de captura del examen:

¿Qué día secuencial del año ocurrió el incidente según los datos del arreglo ya dado? \_148\_\_\_\_\_\_\_\_

Cambia los datos del arreglo por los del arreglo que te aparecerá en la liga de captura y responde de nuevo la pregunta:

¿Qué día secuencial del año ocurrió el incidente según los datos del arreglo ya dado? \_\_\_n/a\_\_\_

***PROBLEMA 3.*** *(30 puntos – Tiempo sugerido para solucionar el problema: 20 min).*

En el archivo EX1.cpp podrás encontrar la definición de una matriz de datos que representa los costos de conexión por hora de los nodos de una red de comunicación de datos (ej. costo de conectar una antena de transmisión de TV con un satélite). La matriz guarda el dato 9999 para representar el valor infinito. Los nodos están identificados del 1 al 7. Adicionalmente, podrás observar que ya se tiene implementado el algoritmo de Floyd. Utiliza este algoritmo para programar las instrucciones necesarias en el *main* y responder las siguientes preguntas en la liga de captura:

¿Cuál es el par de nodos que tiene el costo óptimo (mínimo) de conexión pero que es el más alto (caro) sin importar por cuántos nodos se pase? \_\_\_\_2,1\_\_(valor = 3165)\_\_\_

Leyendo la matriz por renglones y de izquierda a derecha, ¿cuál es el primer par de nodos que no tienen camino de conexión entre ellos? \_\_\_\_segundo con el primero\_\_\_\_\_\_

Cambia los datos del arreglo por los del arreglo que te aparecerá en la liga de captura y responde de nuevo las preguntas:

¿Cuál es el par de nodos que tiene el costo óptimo (mínimo) de conexión pero que es el más alto (caro) sin importar por cuántos nodos se pase? \_\_\_\_n/a\_\_\_\_\_

Leyendo la matriz por renglones y de izquierda a derecha, ¿cuál es el primer par de nodos que no tienen camino de conexión entre ellos? \_\_\_\_n/a\_\_\_\_\_\_

**Una vez que termines el examen, entra a expresatec.mty.itesm.mx y escribe cuál es tu opinión del examen y tu resultado.**

**ALGORITMARIO**

**Algoritmo en C de la búsqueda binaria**

**int busquedaBinaria(int vector[], int n, int dato)**

**{**

**int centro,inf=0,sup=n-1;**

**while(inf<=sup){**

**centro=(sup+inf)/2;**

**if(vector[centro]==dato) return centro;**

**else**

**if(dato < vector [centro]) sup=centro-1;**

**else inf=centro+1;**

**}**

**return -1;**

**}**