

Tecnológico de Monterrey, Campus Monterrey
Departamento de Ciencias Computacionales
ANÁLISIS Y DISEÑO DE ALGORITMOS - Ing. Román Martínez M.
SEGUNDO EXAMEN DE PROGRAMACIÓN - 13 de Noviembre de 2015

Nombre: _____ Matrícula: _____

Instrucciones:

- Este es un examen que podrá ser resuelto desde la comodidad de tu casa, accedando el material del curso y cualquier otra referencia que consideres valiosa para que TÚ generes las respuestas a los problemas planteados.
- El examen deberá resolverse en forma INDIVIDUAL. Evita caer en la tentación de conversar con compañeros del grupo y/o ajenos al grupo. Este examen evalúa TUS conocimientos y habilidades y cualquier acción que realices para entregar un conocimiento o habilidad que no es tuyo, es una acción deshonestas que será penalizada fuertemente de acuerdo a nuestro reglamento. También recuerda que “tanto peca el que mata a la vaca como el que le estira la pata”, así que evita hacerte cómplice de algún compañero en afán de ayudarlo o apoyarlo.
- Para resolver este examen puedes utilizar el compilador de C++ de tu preferencia para programar tus respuestas.
- Cada problema del examen deberá programarse en un archivo propio e independiente, respetando los nombres de archivo que se indican en la redacción del problema.
- Los archivos con los códigos programados se subirán en la página del curso en el espacio indicado en la sección de exámenes.
- Adicionalmente, se te pedirá un video que deberás generar como respuesta a este examen y lo deberás entregar en la página de Facebook del curso a través de un mensaje al *inbox* del profesor. Trata de que este video no exceda de 5 minutos de duración.
- Una vez accesado este examen en la página del curso en Bb, contarás con un tiempo máximo de 5 horas continuas para trabajarlo y entregarlo. El video puede realizarse y entregarse posteriormente a este tiempo de acceso. Considera que tanto los archivos de respuesta como el video, deben estar entregados electrónicamente SIN EXCEPCIÓN antes de la sesión del martes 17 de noviembre a las 9 am. Adicionalmente, en la sesión entregarás estas hojas impresas junto con la impresión de tus códigos.
- Para cualquier duda o aclaración del examen, envía un mensaje por *inbox* al profesor y se te dará respuesta lo más pronto posible.

PROBLEMA 1. (50 puntos).

Realiza la implementación de un programa que le pida al usuario un número entero N como entrada, y sirva para mostrar todas las posibles combinaciones de números enteros positivos que sumados dan como resultado el número N. los números de cada combinación deberán ser todos diferentes y mostrarse en orden siempre ascendente.

Por ejemplo, su N = 6, las combinaciones a mostrar serían:

1+2+3
1+5
2+4
6

Observa que quedan fuera combinaciones como 3+3 y 4+2 por no cumplir las condiciones de números diferentes o estar en orden ascendente.

Para N = 10, las combinaciones a mostrar serían:

1+2+3+4
1+2+7
1+3+6
1+4+5
1+9
2+3+5
2+8
3+7
4+6
10

Para resolver este problema es muy importante que pienses y respondas las siguientes preguntas:

1. ¿Cuál es la técnica de diseño de algoritmos que conviene aplicar para solucionar este problema? ¿qué problema de los expuestos en la técnica te puede servir para solucionar este problema?
2. Dependiendo de la técnica que hayas seleccionado, ¿cuál es el diseño de la solución en términos de la base que requiere la técnica (ej. el árbol de búsqueda de soluciones en Backtracking, o la fórmula recursiva en programación dinámica)?

Como resultado de este problema deberás entregar lo siguiente:

1. El código con la solución implementada y que deberá llamarse P1EX2AD15.cpp. Documenta el código fuente con los comentarios que consideres convenientes para facilitar la revisión del examen, así como con tus datos personales (matrícula, nombre), y el comentario inicial de si tu programa funciona o no. Sube este archivo en la página del curso.
2. En el video deberás mencionar lo siguiente:
 - a. Cómo fue tu proceso de solución al caso, respondiendo a las preguntas de estrategia de solución que se mencionaron anteriormente.
 - b. Si tu programa funciona correctamente y cuál es la solución que da el programa a la prueba donde $N = 10 +$ el último dígito de tu matrícula. Para esto, muestra la pantalla de tu programa en ejecución, y los segmentos de código que consideres importantes explicar.
 - c. Cuánto tiempo tardaste en solucionar el caso y cualquier otro comentario que consideres importante considerar para la revisión.

PROBLEMA 2. (50 puntos).

Muchos de los algoritmos que hemos visto en el curso, requieren seleccionar “algo” entre un conjunto de opciones. Esta selección en el caso de los algoritmos voraces, tiene que ver con la “mejor opción” según la optimización que se quiere lograr (maximizar o minimizar). Dado que esta “mejor opción” se puede relacionar con una **prioridad** entre las opciones que están disponibles, el algoritmo puede plantearse haciendo uso de una cola priorizada o HEAP.

En seguida se presenta un algoritmo voraz (en pseudocódigo) que usa esta idea de una cola priorizada para obtener el árbol de extensión mínima de un grafo no dirigido.

1. Sea A un conjunto de arcos, inicializado como vacío (contendrá la solución del AEM).
2. Sea V un conjunto de nodos vértices, inicializado como un nodo vértice X cualquiera (contendrá los vértices que componen el AEM).
3. Sea CP una cola priorizada, inicializada vacía, y que guardará arcos, siendo el peso del arco la prioridad, y siendo el valor menor el de mayor prioridad.
4. Meter todos los arcos del nodo vértice X en CP.
5. Mientras V no contenga a todos los nodos vértices del grafo:
 - a. Sacar el arco de mayor prioridad de CP
 - b. Asignar a X el nodo vértice que es destino en el arco que se sacó de CP.
 - c. Si el nodo vértice X no está en el conjunto V:
 - i. Agregar el arco que se sacó en el conjunto A.
 - ii. Agregar el nodo vértice X al conjunto V.
 - iii. Meter todos los arcos del nodo vértice X en CP.
6. El árbol de extensión mínima (AEM) lo conforman los arcos contenidos en A.

Realiza la implementación de este algoritmo haciendo uso de la cola priorizada que provee STL. En la siguiente liga: <http://comsci.liu.edu/~jrodriguez/cs631sp08/c++priorityqueue.html> encontrarás una explicación breve y concreta de cómo está definido este tipo de dato en STL y algunos ejemplos que ilustran la forma de usarlo (OJO: el primer ejemplo tiene dos errores que al compilar los podrás detectar y corregir). El programa utilizará como entrada un grafo definido con una matriz de adyacencias y que se declarará con los datos que se te enviarán por correo electrónico. La ejecución del programa mostrará como resultado la lista de arcos que conforman el árbol de extensión mínima del grafo, describiendo cada arco con sus vértices y su valor. Los nodos vértice estarán identificados con los índices de 1 a n. Adicionalmente, el programa indicará cual es el costo mínimo total de conexión de todos los vértices.

Como resultado de este problema deberás entregar lo siguiente:

1. El código con la solución implementada y que deberá llamarse P2EX2AD15.cpp. Documenta el código fuente con los comentarios que consideres convenientes para facilitar la revisión del examen, así como con tus datos personales (matrícula, nombre), y el comentario inicial de si tu programa funciona o no. Incluye también en esta documentación la lista de referencias en Internet que consultaste (si las hubo). Sube este archivo en la página del curso.

2. En el video deberás mencionar lo siguiente:
 - a. ¿Cuál es el algoritmo que corresponde al pseudocódigo que implementaste? Prim, Kruskal u otro
 - b. Cómo fue tu proceso de solución al caso. Qué estructuras de datos utilizaste para representar los conjuntos, y cómo fue tu proceso de entendimiento de la cola priorizada de STL.
 - c. Indica si tu programa funciona correctamente y cuál es la solución que da el programa al grafo que se utilizó para la prueba. Para esto, muestra la pantalla de tu programa en ejecución.
 - d. Cuánto tiempo tardaste en solucionar el caso y cualquier otro comentario que consideres importante considerar para la revisión.

Para cualquier situación no aclarada en este documento, no dudes en preguntarle al profesor.

No olvides que en la sesión del martes 17 de noviembre, deberás entregar estas hojas impresas con tu nombre y engrapadas con la impresión de los códigos que implementaste.