

Indicaciones específicas:

- Esta evaluación contiene 8 páginas (incluyendo esta página) con 5 preguntas. El total de puntos son 20.
- El tiempo límite para la evaluación es 100 minutos.
- Cada pregunta deberá ser respondida en un solo archivo con el número de la pregunta.
 - p1.cpp
 - p2.jpg
 - p3.jpg
 - p4.jpg
 - p5.jpg
- Deberás subir estos archivos directamente a www.gradescope.com, uno en cada ejercicio. También puedes crear un .zip

Calificación:

Tabla de puntos (sólo para uso del professor)

Question	Points	Score
1	5	
2	4	
3	5	
4	3	
5	3	
Total:	20	

1. (5 points) **Patrones de Diseño**

Sistema de Notificación de Calificaciones. Diseña un sistema de notificación de calificaciones para estudiantes utilizando el patrón de diseño "Observer". El sistema debe permitir a los profesores registrar las calificaciones de los estudiantes y notificar automáticamente a los estudiantes interesados cuando sus calificaciones se actualizan.

Requerimientos:

- Crear la clase **Estudiante** que tenga métodos para obtener información básica del estudiante, como nombre, número de identificación, y un método para recibir notificaciones de calificaciones.
- Implementa la clase **Profesor** que tenga un método para registrar las calificaciones de un estudiante.
- Utiliza el patrón de diseño "Observer" creando la interfaz **Observador**. Esta interfaz define el método **actualizarCalificaciones**, que será llamado por el sujeto cuando haya algún cambio en el objeto observado.
- Implemente la clase **SistemaNotificaciones**. Esta clase tiene la responsabilidad de mantener una lista de observadores, proporcionar métodos para agregar y quitar observadores, y notificar a los observadores cuando se produce un cambio. En este caso, el cambio sería una actualización de calificaciones.

Un ejemplo de como debería funcionar el decorador es:

```
int main() {  
    // Crear un estudiante y un profesor  
    Estudiante estudiante("Juan");  
    Profesor profesor;  
  
    // Suscribir al estudiante al sistema de notificaciones  
    profesor.obtenerSistemaNotificaciones().  
        agregarObservador(&estudiante);  
  
    // Registrar calificaciones por parte del profesor  
    profesor.registrarCalificaciones(&estudiante, 85.5);  
  
    return 0;  
}
```

Salida:

```
Estudiante Juan - Calificacion actualizada: 85.5
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Patrones de Diseño	Selección adecuada de los patrones que permiten la solución de algún problema de diseño con una adecuada justificación y descripción a través de un diagrama de clases (5pts).	Selección de un patrón que permite la solución de alguno de los problemas de diseño con una justificación y descripción a través de un diagrama de clases (3pts).	Selección de un patrón que permite la solución de alguno de los problemas de diseño pero no se tiene una justificación y descripción inadecuada del diagrama de clases (2pts).	Selección de un patrón pero no se justificó y descripción incorrecta del diagrama de clases. (1pts)

2. (4 points) Árboles Binarios de Búsqueda

Se implementó un árbol binario de búsqueda insertando números enteros. El siguiente pseudo-código muestra el proceso de creación del árbol:

Listing 1: Creación del Árbol Binario de Búsqueda

```
Crear arbol
arbol.insert(36)
arbol.insert(44)
arbol.insert(21)
arbol.insert(2)
arbol.insert(2)
arbol.insert(7)
arbol.insert(16)
arbol.insert(38)
arbol.insert(10)
arbol.insert(41)
arbol.insert(28)
arbol.insert(20)
arbol.insert(3)
arbol.insert(12)
arbol.insert(1)
arbol.insert(26)
arbol.insert(30)
arbol.insert(22)
arbol.insert(5)
```

Se pide:

- (1 pts) Dibujar el árbol.
- (1 pts) Calcular la altura del árbol.
- (1 pts) ¿En cuantos pasos se encontraría el nodo con clave 1?
- (1 pts) ¿Quién es el ancestro común más bajo para los nodos 30 y 7?

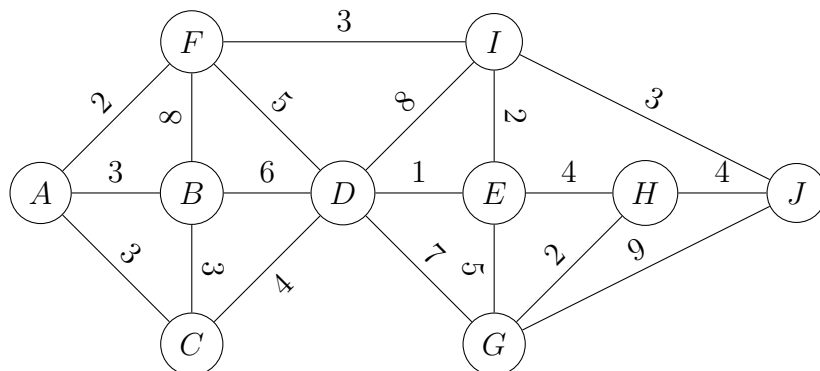
Fundamente su respuesta.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo de Grafos	Descripción detallada, ordenada y fácil de entender. Ilustrado correctamente y sin ningún error aparente. (4pts).	Descripción correcta, ordenada. Ilustrado correctamente y sin ningún error aparente, (3pts)	Descripción correcta, ordenada. Ilustrado correctamente. Con algunos errores, o solución incompleta, o difícil de entender (2pts).	La solución contiene errores. (1pts).

4. (3 points) **Álgoritmo de Kruskal**

Para el siguiente grafo no-dirigido:



Utilizando el algoritmo de Kruskal:

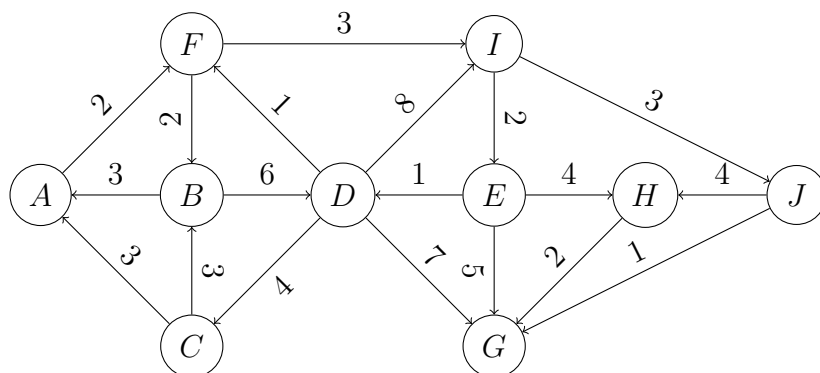
- (3 pts) Cada vez que inserte una arista al Árbol de Expansión Mínimo:
 - (1 pts) Dibuje el árbol generado.
 - (1 pts) Dibuje el Árbol de Expansión Mínimo.
 - (1 pts) Detalle paso a paso como se van generando los árboles. Y finalmente el Árbol de Expansión Mínimo.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo de Grafos	Descripción detallada, ordenada y fácil de entender. Ilustrado correctamente y sin ningún error aparente. (3pts).	Descripción correcta, ordenada. Ilustrado correctamente y sin ningún error aparente, (2pts)	Descripción correcta, ordenada. Ilustrado correctamente. Con algunos errores, o solución incompleta, o difícil de entender (1pts).	La solución contiene errores. (0pts).

5. (3 points) **Álgoritmo Dijkstra**

Para el siguiente grafo dirigido:



Utilizando el algoritmo de Dijkstra. Describa las distancias para cada nodo utilizando como origen el nodo D .

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo de Grafos	Descripción detallada, ordenada y fácil de entender. Ilustrado correctamente y sin ningún error aparente. (3pts).	Descripción correcta, ordenada. Ilustrado correctamente y sin ningún error aparente. (2pts)	Descripción correcta, ordenada. Ilustrado correctamente. Con algunos errores, o solución incompleta, o difícil de entender. (1pts).	La solución contiene errores. 0 pts).