

Indicaciones específicas:

- Esta evaluación contiene 7 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.cpp
 - 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	4	
2	4	
3	4	
4	4	
5	4	
Total:	20	

1. (4 points) **Patrones de Diseño**

La clase `Fibonacci` permite calcular el término `n` en la serie de *Fibonacci*. Utilice un *Decorador* para que un objeto de tipo `Fibonacci` ahora también imprima el tiempo requerido para esta operación.

Un ejemplo de como debería funcionar el decorador es:

```
int main(){
    Fibonacci f;
    int f20 = f(20);
    cout << "Fibonacci(20) = " << f20 << endl;

    FibonacciTiempo ft(&f);
    int ft20 = ft(20);           // Al calcular el valor de
                                // ft20 se debe imprimir el
                                // tiempo consumido para esta
                                // operacion.
    cout << "Fibonacci(20) = " << ft20 << endl;

    return 0;
}
```

Salida:

```
Fibonacci(20) = 55

Tiempo: 0.055ms
Fibonacci(20) = 55
```

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Implementación del Algoritmo	La implementación del algoritmo es clara y ordenada. El desarrollo de la solución es detallado y correcto (4pts).	La implementación del algoritmo es clara y ordenada. El desarrollo de la solución es suficiente (3pts)	La solución es parcialmente correcta (2pts).	La solución es incorrecta(1pts)

2. (4 points) **Pilas y Colas**

Se pide implementar una nueva estructura **MinMaxStack**, la cual tenga las siguientes operaciones:

- **push**: Insertar un elemento en la parte superior (igual que un **STACK**). Tiempo computacional: $\mathcal{O}(1)$.
- **top**: Retornar el elemento en la parte superior (igual que un **STACK**). Tiempo computacional: $\mathcal{O}(1)$.
- **pop**: Eliminar el elemento en la parte superior (igual que un **STACK**). Tiempo Computacional: $\mathcal{O}(1)$.
- **min**: Retornar el mínimo valor insertado hasta el momento. Tiempo computacional: $\mathcal{O}(1)$.
- **max**: Retornar el máximo valor insertado hasta el momento. Tiempo computacional: $\mathcal{O}(1)$.

Utilice la función **main()** para probar todas estas operaciones. Si no se implementan las operaciones **min** y **max** la nota será CERO.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Adaptadores	Muy buen nivel de abstracción, código simple de leer y entender, pocas líneas de código, considerando la eficiencia, uso adecuado de estructuras y selección correcta de los tipos o template. (4pts).	Adecuado nivel de abstracción, uso adecuado de estructura y selección adecuada de los tipos o template. Pero con ciertas limitaciones al generalizarlo. (3pts)	La solución es parcialmente correcta (2pts).	La solución es incorrecta(1pts)

3. (4 points) Árboles Binarios de Búsqueda

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

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 7?
- (1 pts) ¿El árbol esta balanceado?

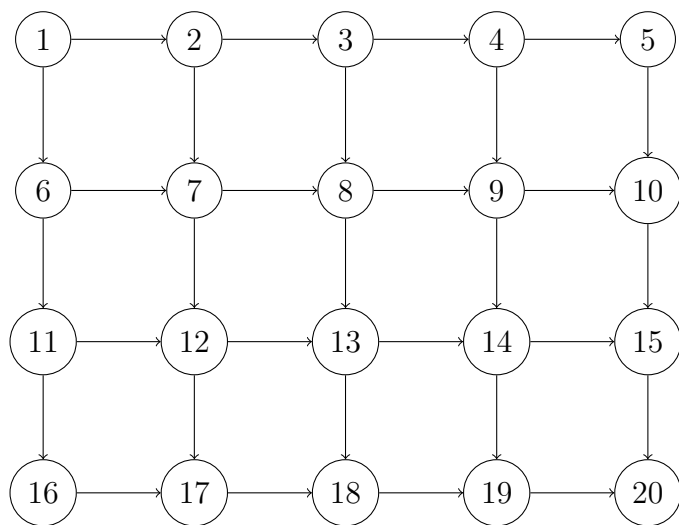
Fundamente su respuesta.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo de Grafos	Muy buen detalle de la descripción, ordenada y fácil de leer, ilustrado correctamente y sin ningún error aparente, muy ordenado. (4pts).	Buen detalle de la descripción, ilustrado correctamente y sin ningún error aparente, ordenado. (3pts)	Buen detalle de la descripción, ilustrado correctamente y con ligeros error o incompletos o difícil de entender (2pts).	Contiene errores que no se hace lo solicitado (1pts).

4. (4 points) **Algoritmo DFS**

En el siguiente grafo dirigido, el algoritmo DFS se aplica de una manera especial. Para un nodo dado, se visita solamente un nodo vecino, escogido de manera aleatoria.



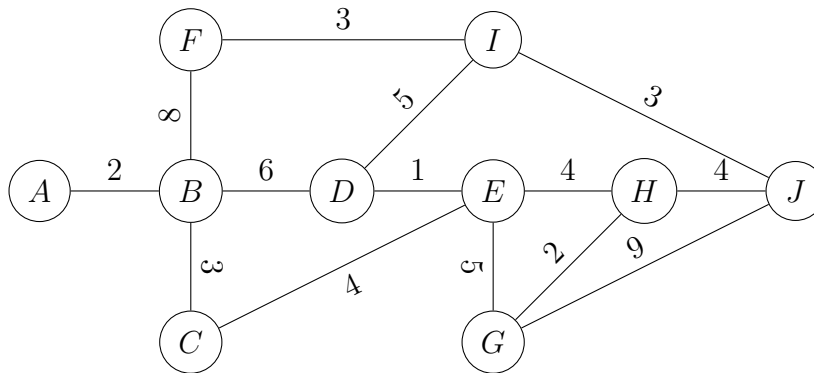
Indique la secuencia de nodos visitados utilizando esta variante del algoritmo DFS, utilice como origen el nodo con clave 1. Además, marque la arista utilizada cada vez que un nodo visita su vecino aleatorio.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo de Grafos	Muy buen detalle de la descripción, ordenada y fácil de leer, ilustrado correctamente y sin ningún error aparente, muy ordenado. (4pts).	Buen detalle de la descripción, ilustrado correctamente y sin ningún error aparente, ordenado. (3pts)	Buen detalle de la descripción, ilustrado correctamente y con ligeros error o incompletos o difícil de entender (2pts).	Contiene errores que no se hace lo solicitado (1pts).

5. (4 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:
 - Dibuje el árbol generado.
 - Detalle como se unieron los ancestros del par de vértices.
 - Finalmente, indique la raíz del árbol.
- (1 pts) Al terminar el algoritmo, dibuje el Árbol de Expansión Mínimo. Para ello, únicamente marque las aristas del grafo.

La rúbrica para esta pregunta es:

Criterio	Excelente	Adecuado	Mínimo	Insuficiente
Algoritmo de Grafos	Muy buen detalle de la descripción, ordenada y fácil de leer, ilustrado correctamente y sin ningún error aparente, muy ordenado. (4pts).	Buen detalle de la descripción, ilustrado correctamente y sin ningún error aparente, ordenado. (3pts)	Buen detalle de la descripción, ilustrado correctamente y con ligeros error o incompletos o difícil de entender (2pts).	Contiene errores que no se hace lo solicitado (1pts).