

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**ALGORITMIA Y ESTRUCTURA DE DATOS**  
**Primer Examen**  
**(Segundo Semestre 2024)**

Duración: 2h 50 min.

- **No puede utilizar apuntes, solo hojas sueltas en blanco.**
- En cada función el alumno deberá incluir, a modo de comentario, la forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear plantillas o funciones no vistas en los cursos de programación de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado. El orden será parte de la evaluación.
- **Solo está permitido acceder a la plataforma de PAIDEIA, cualquier tipo de navegación, búsqueda o uso de herramientas de comunicación se considera plagio por tal motivo se anulará la evaluación y se procederá con las medidas disciplinarias dispuestas por la FCI.**
- Para esta evaluación solo se permite el uso de las librerías `iostream`, `iomanip`, `limits`, `string`, `cmath` o `fstream`
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_EX1_P#` (donde # representa el número de la pregunta a resolver)

---

**Pregunta 1 (10 puntos)**

Un almacén coloca sus productos apilados uno sobre otro, luego las pilas formadas son asignadas una al lado de la otra, formando una larga fila de pilas. Esta forma de almacenamiento muchas veces quita la visibilidad de los productos que se encuentran tras una pila muy alta, lo cual dificulta el manejo de los productos dentro del almacén. Por tal motivo la empresa ha adquirido un robot dron que, al visualizar la altura de cada pila, puede determinar aquella que es mas alta que las pilas que están a su alrededor. Como pueden existir pilas de la misma altura se desea que robot dron sea capaz de determinar aquella pila que tiene mayor cantidad de pilas mas pequeñas a su alrededor. A continuación, algunos ejemplos:

Si el robot detecta una fila con las siguientes alturas: 1 3 6 4

**Las respuestas serán: Posición=2, 3 pilas pequeñas alrededor**

La explicación de estos valores es que la pila de la posición 0, no tiene ninguna pila de menor altura a su alrededor, la pila de la posición 1 tiene una pila mas pequeña alrededor, la pila de la posición 2, tiene 2 pilas al lado izquierdo de menor altura y una pila al lado derecho de menor altura con lo que suma 3 y la pila de la posición 3 no tiene ninguna pila de menor altura a su alrededor.

Si el robot detecta una fila con las siguientes alturas: 2 5 7 7 1

**Las respuestas serán: Posición=2, 2 pilas pequeñas alrededor**

La explicación de estos valores es que la pila de la posición 0, no tiene ninguna pila más pequeña a su alrededor, la pila de la posición 1 tiene una pila más pequeña alrededor, la pila de la posición 2 tiene 2 pilas al lado izquierdo de menor altura lo que suma 2, la pila de la posición 3 tiene una pila de menor tamaño al lado derecho, la pila de la posición 4 no tiene ninguna pila de menor altura a su alrededor.

Si el robot detecta una fila con las siguientes alturas: 1 7 2 7 3 4 3 2 1 7 2 1 7 3

**Las respuestas serán: Posición=9, con 7 pilas pequeñas alrededor**

Si el robot detecta una fila con las siguientes alturas: 6 2 5 4 5 1 6

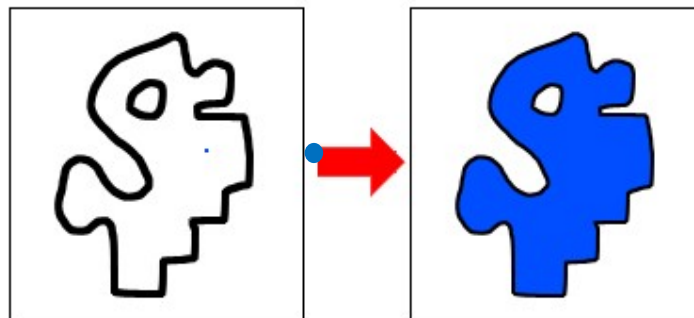
**Las respuestas serán: Posición=0, con 5 pilas pequeñas alrededor**

En este caso también sería correcta la posición 6

Desarrolle un programa de complejidad  $O(n)$  que utilizando una **pila auxiliar** determine la posición de la pila más alta y la cantidad de pilas de menor altura a su alrededor. Los datos de las alturas de las pilas se reciben en un arreglo. **Para que la respuesta sea válida debe determinar ambos resultados solicitados.**

## Pregunta 2 (10 puntos)

- a. **(5.0 puntos)** Imagine que trabaja en un software de edición de imágenes. Una de las características clave de este software es la herramienta de relleno, que permite a los usuarios cambiar el color de una región específica de la imagen, similar a la herramienta de "pintura" que se encuentra en muchas aplicaciones gráficas. La herramienta debe ser capaz de rellenar áreas contiguas del mismo color con un nuevo color, independientemente de la forma de la región.

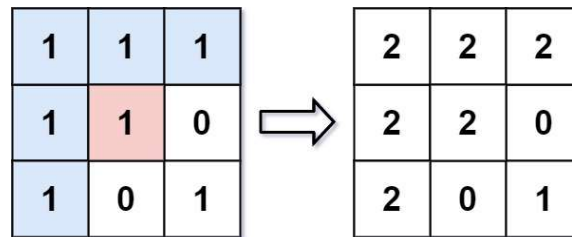


Para resolver este problema tener en cuenta lo siguiente:

1. La imagen es una matriz 2D donde cada celda contiene un valor que representa el color de ese píxel.
2. El usuario hace clic en un píxel en la posición (x, y) que tiene un color específico (por ejemplo, el valor 1).
3. El usuario elige un nuevo color (por ejemplo, el valor 2).
4. El algoritmo debe buscar **recursivamente** todos los píxeles contiguos que tienen el color 1 y cambiarlos a 2. **No considere los píxeles en posiciones diagonales.**

A continuación, una imagen de ejemplo:

Posición del píxel= (1,1) nuevo valor = 2



Se le pide:

- a) Implementar las estructuras necesarias, así como el ingreso de datos, considerando como entrada una **matriz numérica** de cualquier dimensión, la posición del pixel en (x, y), y el valor del nuevo color.
- b) Implementar una **función recursiva** para pintar todos los pixeles afectados usando movimientos arriba, abajo, izquierda y derecha.

**No puede emplear memoria extra o arreglos adicionales a la matriz de entrada, tampoco puede usar TADs adicionales (listas, pilas, colas).**

- b. (5 puntos) Se tiene un sistema de atención de clientes diseñado para un banco. Donde los clientes llegan cada **15 segundos** y se atienden con un cajero como máximo. El sistema se maneja con una cola de prioridad, en la que los clientes VIP (cada 2 clientes vienen un cliente VIP) tienen prioridad para ser atendidos antes que los demás. Si un cliente VIP espera más de 30 segundos en la cola, se retira sin ser atendido. El proceso dura 20 segundos. Tu tarea es simular este sistema para un cajero y clientes (6) y responder las siguientes preguntas:

**Calcular los resultados de la simulación:**

- Clientes VIP fueron atendidos exitosamente.
- Clientes VIP abandonaron la cola debido a esperas largas (más de 30 segundos).

**Asumiendo que los clientes VIP llegan cada 2 clientes:**

- **Cliente 1 (no VIP):** Llega en el segundo 15, es atendido inmediatamente. Sale en el segundo 35.
- **Cliente 2 (VIP):** Llega en el segundo 30. Se inserta en la cola de VIP.
- **Cliente 3 (no VIP):** Llega en el segundo 45. Se inserta al final de la cola de no VIP.
- **Cliente 4 (VIP):** Llega en el segundo 60. Se inserta **antes de Cliente 3** en la cola de VIP.
- **Cliente 5 (no VIP):** Llega en el segundo 75. Se inserta al final de la cola de no VIP.
- **Cliente 6 (VIP):** Llega en el segundo 90. Se inserta en la cola de VIP.

**Flujo de atención:**

1. **Cliente 1 (no VIP):** Llega a los 15 segundos y es atendido inmediatamente. Sale a los 35 segundos.

2. **Cliente 2 (VIP):** Llega a los 30 segundos. Es atendido después de **Cliente 1**. Sale a los 55 segundos.
3. **Cliente 3 (no VIP):** Llega a los 45 segundos. Se encola y espera.
4. **Cliente 4 (VIP):** Llega a los 60 segundos. Se inserta en la cola de VIP, pero espera porque el cajero está ocupado.
5. **Cliente 5 (no VIP):** Llega a los 75 segundos. Se inserta al final de la cola de no VIP.
6. **Cliente 6 (VIP):** Llega a los 90 segundos. Se inserta en la cola de VIP, pero espera porque el cajero está ocupado.

**Resultados esperados para 6 clientes:**

- **Clientes VIP atendidos exitosamente:** 3 (Clientes 2, 4 y 6).
- **Clientes VIP no atendidos:** 0.

RESULTADOS DE LA SIMULACIÓN DEL DESPACHO DE ENTRADAS	
Clientes VIP atendidos	Clientes VIP no atendidos
3	0

**Para la solución de esta pregunta solo puede usar una cola de prioridad, no puede emplear arreglos ni TADs adicionales. Se le sugiere emplear la implementación de cola con prioridad que se encuentra en Paideia.**

### Pregunta 3 (10 puntos)

- a. **(5.0 puntos)** Una empresa que maneja y gestiona información crítica desea implementar una verificación de ataque de tipo "escalada de privilegios" en su red de computadoras para detectar posibles vulnerabilidades. La red está configurada de manera que se tiene registrada la conectividad entre las computadoras, es decir, si una computadora tiene permiso para acceder a otra.

La red puede modelarse como un arreglo **de listas**:

- El arreglo principal representa todas las computadoras de la empresa (hay **n** computadoras en total).
- Cada elemento del arreglo principal contiene una lista que almacena las computadoras a las que esa computadora específica tiene permiso para acceder (hay **m** conexiones en total en toda la red).

Se solicita implementar un programa en **C++** que simule esta red de conectividad utilizando exclusivamente **listas** como estructura de datos. El objetivo es determinar en un momento dado si, dada una computadora origen y una computadora destino, existe un camino (directo o a través de otras computadoras) que permita el acceso de la computadora origen a la computadora destino, simulando así si un atacante podría escalar sus privilegios para acceder a información confidencial.

Considere la siguiente actividad en la red para probar su solución:

```
X
Se conectó la computadora 1 con la computadora 2
Se conectó la computadora 1 con la computadora 3
Se conectó la computadora 2 con la computadora 4
Se conectó la computadora 5 con la computadora 6

Computadora 1: -> 2 -> 3
Computadora 2: -> 4
Computadora 3:
Computadora 4:
Computadora 5: -> 6
Computadora 6:

VERIFICACIÓN: de computadora 1 a 6
Resultado: No hay escalada de privilegios

Se conectó la computadora 3 con la computadora 5
Se desconectó la computadora 5 con la computadora 6

Computadora 1: -> 2 -> 3
Computadora 2: -> 4
Computadora 3: -> 5
Computadora 4:
Computadora 5:
Computadora 6:

VERIFICACIÓN: de computadora 1 a 6
Resultado: No hay escalada de privilegios

Se conectó la computadora 4 con la computadora 6

Computadora 1: -> 2 -> 3
Computadora 2: -> 4
Computadora 3: -> 5
Computadora 4: -> 6
Computadora 5:
Computadora 6:

VERIFICACIÓN: de computadora 1 a 6
Resultado: Si hay escalada de privilegios
```

A partir de la implementación de Listas realizada en el curso (descárguela de PAIDEIA):

1. Modifique las estructuras y funciones para gestionar el arreglo de listas. Desde el **main()** inserte los datos al arreglo de listas de manera directa.
2. Implemente la función **agregarConexion()**, que permita agregar una conexión entre una computadora origen y una computadora destino.
3. Implemente la función **eliminarConexion()**, que permita eliminar una conexión entre una computadora origen y una computadora destino.
4. Implemente la función **hayEscaladaPrivilegios()**, que verifique si existe un camino desde la computadora origen a la computadora destino. La complejidad no debe superar **O(n+m)**.
5. Implemente la función **imprimirRed()**, que muestre las computadoras y sus conexiones.

- b. (5 puntos) Un número **vampiro** es un número natural con un número par de dígitos, igual al producto de dos números naturales, donde cada uno de los multiplicandos posee la mitad de dígitos que el original. Además, estos números contienen entre ambos, todos los dígitos del número inicial.

**Por ejemplo**, el número 1260 es un número vampiro, porque tiene 4 dígitos (número par de dígitos) y es el resultado de multiplicar 21 (2 dígitos) x 60 (2 dígitos) siendo los dígitos de los números 21 y 60 exactamente los 4 dígitos del número 1260.

Otro número vampiro es 136948, porque tiene 6 dígitos y es el resultado de multiplicar 146 (3 dígitos) x 938 (3 dígitos), siendo los dígitos de los números 146 y 938 exactamente los 6 dígitos del número 136948. Con lo anterior explicado, se le pide lo siguiente:

Implementar una función **verificaVampiro** que permita verificar si un número es vampiro o no. Para ello debe evaluar si existe alguna posible combinación de números, formados con los dígitos del número original que cumplan con lo indicado en el enunciado. Para esta evaluación debe utilizar la estrategia de **fuerza bruta**, es decir, debe evaluar todas las posibles combinaciones que se pueden formar con los dígitos y si una de ellas cumple, entonces es **vampiro**. (5 puntos).

Al finalizar el examen, comprima la carpeta de su proyecto empleando el programa Zip que viene por defecto en el Windows, **no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares**. Luego súbalo a la tarea programa en Paideia para este examen.

Profesores del curso:

Ana Roncal  
Fernando Huamán  
David Allasi  
Heider Sanchez  
Rony Cueva

San Miguel, 19 de octubre del 2024