

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

**ALGORITMIA**

**Segundo Examen**

**(Segundo Semestre 2023)**

Duración: 2h 50 min.

- En cada función el alumno deberá incluir, a modo de comentario, la estrategia o 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 STL, Plantillas o funciones no vistas en los cursos 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.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- **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.**
- **Los programas deben ser desarrollados utilizando nombres para las funciones y variables en español, al igual que los comentarios. El uso de otro idioma anula su respuesta.**
- Para esta evaluación solo se permite el uso de las librerías **iostream, iomanip, climits cmath, fstream y cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- **Es obligatorio usar como compilador NetBeans. Solo pueden usar una biblioteca para cada tipo de TAD, cualquier duplicidad o unión de bibliotecas invalidará su respuesta.**
- Los archivos deben llevar como nombre su código de la siguiente forma `codigo_EX2_P#` (donde # representa el número de la pregunta a resolver)

---

**Resuelva solo 2 de las siguientes preguntas:**

**Pregunta 1 (10 puntos)**

Durante un conflicto en el medio oriente, una de las naciones en disputa, ha decidido enviar sus mensajes con las coordenadas de importancia bélica, mezcladas con números que al juntarlos forman números capicúas, con el fin que su oponente, que es un aliado de la OTLAN, se confunda y así no puedan descifrar sus mensajes a pesar de que cuentan con una gran red de espionaje que logra interceptar sus comunicaciones

Luego de un tiempo el aliado de la OTLAN descubre la estrategia de la nación del medio oriente, así que decide dividir el mensaje separando las cifras que forman números capicúas. A continuación, se muestra un ejemplo:

Si la nación del medio oriente envía el siguiente mensaje, con coordenadas ocultas:

**10 20 20 10 11 23 13 23 30 25**

El mensaje se puede dividir realizando 4 cortes, quedando como coordenadas válidas los números 11, 30 y 25.

Otro ejemplo:

16 22 22 52 11 11 11 4 28 19 28 28

El mensaje se puede dividir realizando 6 cortes, quedando como coordenadas válidas los números 16, 52, 4 y 28.

El aliado de la OTLAN requiere que desarrolle un algoritmo que usando programación dinámica calcule con cuantos cortes puede hallar las coordenadas validas dentro de un mensaje enviado. Para responder esta pregunta puede emplear un arreglo o matriz de apoyo a la tabla de respuestas parciales. No es necesario emplear un ingreso de datos.

**Nota: Para la pregunta es obligatorio mostrar la tabla (arreglo o matriz) DP o F, con las respuestas parciales, de no hacerlo la respuesta no será calificada. Recuerda esta tabla debe considerar las respuestas parciales exactas.**

## Pregunta 2 (10 puntos)

Las tortuninjas ingresaron a la pizzería Freddy Fazbear's Pizza para celebrar su triunfo sobre SuperFly, sin embargo, fueron capturados por un zorro animatrónico rojo, quien los toma prisioneros. En un intento de escapar, Donatello prepara un árbol binario de letras, donde envía un mensaje oculto al maestro Splinter. El árbol binario enviado al maestro Splinter es similar al de la figura 1. Adicionalmente al árbol, Donatello, logra remitirle un papel con un conjunto de palabras, dentro de las cuáles se encuentra el mensaje oculto registrado en el árbol binario.

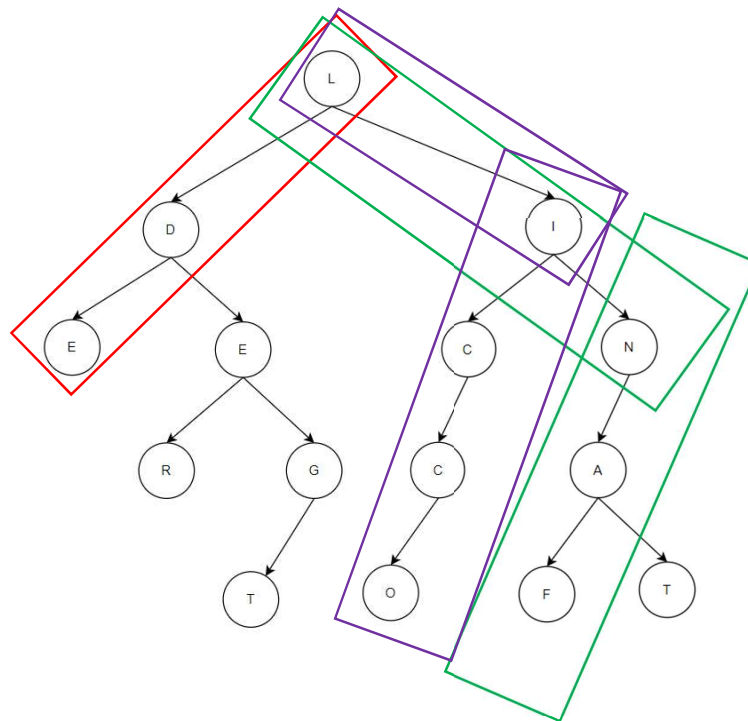


Figura 1. Árbol binario enviado al maestro Splinter.

Las palabras que forman parte del mensaje oculto son aquellas que se pueden formar con todas las letras que están en los caminos que van desde la raíz a cada una de las hojas. Por ejemplo, Donatello envió en el papel las siguientes palabras: HIJO, FINAL, MUNDO, DEL, DIA, CICLO.

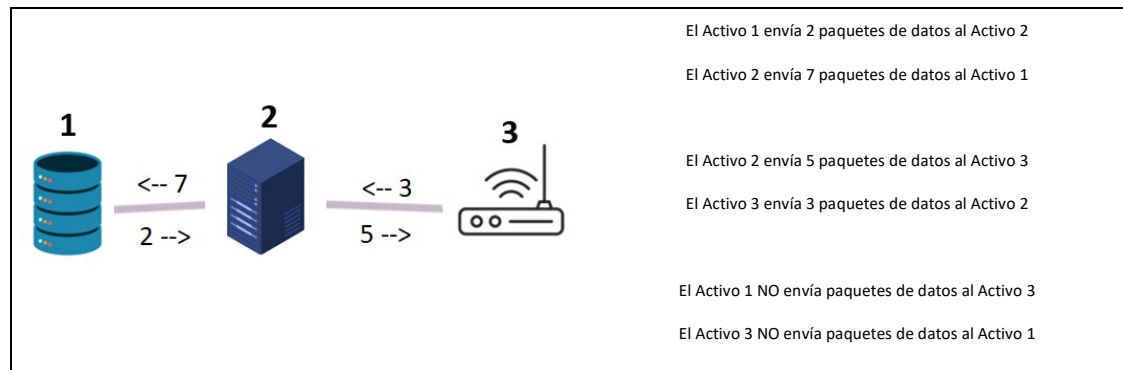
Si se revisa en el árbol binario, la palabra **FINAL** se encuentra en el árbol en el camino de color **Verde**, la palabra **DEL** se encuentra en el camino de color **Rojo** y la palabra **CICLO** se encuentra en el camino de color **Morado**. Las demás palabras del papel no se encuentran en el árbol, porque no se pueden formar al recorrerlo, por lo tanto, el mensaje oculto es **FINAL DEL CICLO**.

- Elabore una función que permita indicar si existe una palabra dentro del árbol binario. La palabra debe ser recibida como un arreglo de caracteres y es el único arreglo que podrá utilizar, en caso desee utilizar alguna estructura adicional, solo podrá ser una lista simplemente enlazada (8 puntos).
- Elabore una función main que permita procesar todas las palabras ingresadas e imprima el mensaje oculto. **Esta pregunta solo tiene valor si ha realizado la parte anterior** (2 puntos).

### Pregunta 3 (10 puntos)

Uno de los mayores retos en seguridad de la información es la Gestión de Accesos a los activos de información de la empresa. Para efectos prácticos se podría decir que estos accesos se brindan considerando la de cantidad de paquetes de que se envían entre los activos sobre la arquitectura de la empresa.

Por ejemplo, del siguiente esquema se entiende que:



Si se considera el siguiente esquema de paquetes enviados entre los activos de información de una empresa:



Matriz de paquetes enviados entre los activos:

	1	2	3	4	5	6	7	8	9
1	1	2	0	6	0	0	0	0	2
2	7	1	5	2	0	0	0	0	7
3	0	3	1	2	3	8	9	7	5
4	2	7	5	1	2	0	0	0	0
5	0	0	5	3	1	4	0	0	0
6	0	0	9	0	6	1	2	0	0
7	0	0	3	0	0	7	1	4	0
8	0	0	4	0	0	0	9	1	9
9	5	3	6	0	0	0	0	4	1

- El valor 0 indica que no se envían paquetes.
- El valor en la celda significa la cantidad de paquetes (de 2 a 9) que envía un Activo a otro.
- El valor 1 indica que el Activo está habilitado

Lo que se busca es “etiquetar” cada activo con un número para asegurar que 2 activos que se envían paquetes de datos no tengan la misma etiqueta. Este etiquetado es “guardado” para poder identificar en el futuro cambios no autorizados en los envíos de paquetes datos. Si en algún momento 2 Activos que se envían paquetes de datos tienen el mismo etiquetado, significa que ha habido una vulneración en la arquitectura de seguridad de la empresa.

A continuación, se muestra el etiquetado que se realizará a los activos mostrados como ejemplo:



Cantidad de tipos de etiquetas: 4

La solución sería:

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

Cantidad de tipos de etiquetas: 3

No habría solución. Ya que no se podría cumplir que 2 activos que se envían paquetes de datos tengan etiquetas distintas.

Se le solicita:

Implemente un algoritmo, utilizando **backtracking**, que permita generar el etiquetado para los activos de información. Los datos de entrada mínimos son (i) la matriz que representa la cantidad de paquetes de datos que se envían entre los activos y (ii) la cantidad de tipos de etiquetas, los cuáles los ingresará directamente en el main(). La salida debe ser las etiquetas asignadas a cada activo o, en caso no se pueda cumplir con que todos los activos conectados tengan etiquetas distintas, mostrar que no hay solución. (10 puntos).

Datos de entrada: (mínimos, usted puede utilizar más si desea)

**Matriz brindada en el ejemplo:**

```
{1, 2, 0, 6, 0, 0, 0, 2},  
{7, 1, 5, 2, 0, 0, 0, 7},  
{0, 3, 1, 2, 3, 8, 9, 7, 5},  
{2, 7, 5, 1, 2, 0, 0, 0, 0},  
{0, 0, 5, 3, 1, 4, 0, 0, 0},  
{0, 0, 9, 0, 6, 1, 2, 0, 0},  
{0, 0, 3, 0, 0, 7, 1, 4, 0},  
{0, 0, 4, 0, 0, 0, 9, 1, 9},  
{5, 3, 6, 0, 0, 0, 0, 4, 1}}
```

**Tipos de etiquetas:** 3 (luego pruebe con 4).

Profesores del curso:

David Allasi  
Fernando Huamán  
Rony Cueva

San Miguel, 02 de diciembre del 2023