



Instituto Tecnológico y de Estudios Superiores de Monterrey

Análisis y diseño de algoritmos avanzados (Gpo 601) – TC2038.601

Tutor Professor: Ramona Fuentes Valdéz

Actividad Integradora 1

Juan Eduardo Rosas Cerón - A01710168

Juan Carlos Calderón García - A01625696

Pablo Hazael Hurtado Mireles - A01710778

2/10/2024

# Problemática

Cuando se transmite información de un dispositivo a otro, se transmite una serie sucesiva de bits, que llevan una cabecera, datos y cola. Existe mucha gente mal intencionada, que puede interceptar estas transmisiones, modificar estas partes del envío, y enviarlas al destinatario, incrustando sus propios scripts o pequeños programas que pueden tomar cierto control del dispositivo que recibe la información

Suponiendo que conocemos secuencias de bits de código mal intencionado:

- ¿serías capaz de identificarlo dentro del flujo de bits de una transmisión?
- ¿podremos identificar si el inicio de los datos se encuentra más adelante en el flujo de bits?

Si tuviéramos dos transmisiones de información y sospechamos que en ambas han sido intervenidas y que traen el mismo código malicioso, ¿podríamos dar propuestas del código mal intencionado?

Para resolver esto se nos proporcionaron 2 transmisiones y 3 patrones de código malicioso, y con estos debemos lograr 3 cosas:

1. Identificar si en las transmisiones hay código malicioso
2. Encontrar el palíndromo más grande de cada transmisión
3. Encontrar la subcadena continua común más grande de ambas transmisiones

## Propuestas de solución

### Parte 1: Algoritmo Z

Complejidad  $O(n)$ . El algoritmo Z lo utilizamos principalmente para comparar si existe código malicioso dentro de los archivos de transmisión, este algoritmo es muy eficiente ya que va guardando las posibles respuestas en un arreglo Z, para después compararlo con el código malicioso. Esto si lo comparamos al tratar de responder a fuerza bruta la complejidad escalaría enormemente.

### Parte 2: Algoritmo Manacher

Complejidad  $O(n)$ . El algoritmo de Manacher logra encontrar el palíndromo en tiempo lineal y ya que los strings de las transmisiones pueden llegar a ser de cerca de 200 caracteres utilizar algo con una complejidad más alta que una lineal haría que el programa fuera mucho más lento.

### Parte 3: Algoritmo Longest Common Subsequence - LCS

Complejidad  $O(m * n)$  donde  $m$  y  $n$  son las longitudes de los textos de entrada. El algoritmo Longest Common Subsequence (LCS) es útil cuando deseas identificar la secuencia más larga que aparece en el mismo orden en dos cadenas, aunque no de forma continua. Este algoritmo logra detectar un mismo código malicioso en dos transmisiones y es efectivo porque permite identificar patrones de código que pueden estar dispersos o ligeramente modificados en ambas transmisiones. A diferencia de encontrar una subcadena común exacta, LCS permite detectar patrones dispersos en los datos, lo que lo hace más flexible y robusto para este tipo de análisis.

#### Conclusiones finales:

Nuestra conclusión radicó en cómo los algoritmos que manejamos nos ayudan para crear código más eficiente, en cómo estos reducen enormemente las iteraciones necesarias para realizar las mismas acciones. Esto no solo mejora nuestras habilidades en programación competitiva, sino que también nos ayuda a mejorar como profesionales y en muchas áreas de la propia carrera.

#### Referencias:

GeeksforGeeks. (2021, 11 febrero). *Manacher's Algorithm Linear Time Longest Palindromic*

*Substring part 1*. GeeksforGeeks.

<https://www.geeksforgeeks.org/manachers-algorithm-linear-time-longest-palindromic-substring-part-1/>

GeeksforGeeks. (2024, 16 julio). *Z algorithm (Linear time pattern searching Algorithm)*.

GeeksforGeeks.

<https://www.geeksforgeeks.org/z-algorithm-linear-time-pattern-searching-algorithm/>

*JavaScript interview with a Google engineer*. (2023, June 27). [Video].

<https://interviewing.io/questions/longest-common-subsequence>