

Problema C

0. Identificación:

Autores del Problema C

- Tony Santiago Montes Buitrago [202014562] <t.montes@uniandes.edu.co>
- Juan Carlos Marin Morales [202013973] <j.marinm@uniandes.edu.co>

1. Algoritmo de Solución:

Para la solución de este problema se propuso una solución que tiende a ser de fuerza bruta, ya que analiza todos los posibles casos para obtener la cadena de menor longitud entre todas las posibles utilizando como superconjunto a todas las posibles permutaciones de sí mismo del conjunto de palabras.

Sobre cada permutación se realizó recursivamente un overlap (una cadena sobre la otra, con la mínima longitud) de cada elemento sobre el overlap de la anterior, hasta completar toda la cadena, y así obtener una palabra descifrada, pero no necesariamente con longitud mínima.

Para hallar la cadena con longitud mínima se realizó al final un min entre todas las palabras obtenidas, con el parámetro llave que retorna la longitud de la cadena (que es lo que se quiere minimizar).

Métodos:

Para mayor manejo matemático, se modelan las cadenas (strings) como listas de caracteres (char).

1. overlap

Precondición: $Q_1: \{ |w_1| = |w_2| \}$

Postcondición: “Toda posible cadena que contenga todas las palabras tiene longitud mayor o igual a w ”

$ismin(w, words) \equiv (\forall string | (\forall s: words |: s \subseteq string): |w| \leq |string|) \wedge (\forall s: words |: s \subseteq w)$

$R_1: \{ c = (+ char : w_1 | char \in w_2 : 1) \wedge ismin(w, [w_1, w_2]) \}$

2. min_str

Precondición: $Q_2: \{ (\forall \langle i, j \rangle | 0 \leq i, j < |words| : |words[i]| = |words[j]|) \}$

Postcondición: $R_2: \{ ismin(w, words) \}$

2. Análisis de Complejidad:

a. Temporal: $O(n! \cdot n \cdot k)$

Se recorren todas las permutaciones, sin embargo, el número de permutaciones es $n!$; sobre cada iteración se revisan $n - 1$ veces y se halla el overlap entre cada par de elementos. El overlap tiene una complejidad de k , por ende se multiplican todas estas variables.

b. Espacial: $O(1)$

La única “lista” que se utilizó en este caso fue el generador de todas las permutaciones, sin embargo, al ser un generador, no utiliza el espacio en memoria que utilizaría una lista; sino que utiliza únicamente un espacio a la vez.

Comentarios Finales:

El desempeño de esta solución, al contrario de las de los problemas A y B, es mucho peor, ya que $n!$ tiende a crecer mucho cuando el valor de n es mayor a 7; es por esto que a largo plazo esta solución no es muy sostenible, sin embargo, para valores de $n \leq 6$ se logra proporcionar una solución en un tiempo no muy extenso.