

# CONSTRUYENDO PUEBLOS

AUTOR: LAUTARO LASORSA

## Solución Esperada

Para la solución de este problema es necesario el uso de la estructura **Trie**<sup>1</sup> (para esto observar que, en este caso, las posiciones de un vector de enteros pueden tratarse de igual forma que los caracteres de un string)

Como todas los caminos tienen el mismo largo, son todos hojas en el **Trie**, lo que da una mayor comodidad para la solución. En esta solución nos va a interesar agregar una información a cada nodo, que es la suma del valor en las hojas de su subárbol. Es decir, teniendo en cuenta que cada nodo representa un prefijo de uno o más de los caminos que se insertaron en el **Trie**, la suma de los valores asociados a los caminos de los cuales es prefijo.

Este se puede actualizar fácilmente, ya que para llegar a un nodo recorreremos todos sus ancestros en el **Trie**. Por lo tanto, lo único que hay que hacer es modificar el valor de cada uno de esos ancestros cuando los visitamos. Esta observación es suficiente para responder los llamados a la función **Cambiar**, y una parte de la solución a los llamados de la función **Crear**.

Teniendo el **Trie**, la pregunta más difícil pasa a ser como responder rápidamente (en un tiempo lineal en el largo de los strings) la pregunta : ¿Dadas 2 hojas (que se crean en este mismo paso), cuál es la suma de los valores de las hojas que hay en el medio?

Esto se debe responder recorriendo el Trie simultáneamente en los 2 caminos relacionados a las hojas de la pregunta. Siendo **posa** la ubicación del puntero que recorre el camino del string **A**, y **posb** el relacionado al string **B**, existen 3 posibilidades para cada **i** ( $0 \leq i < M$ ) :

- **posa = posb** y **A[i] = B[i]**. Indica que los primeros **i** pasos de ambos caminos son iguales, no debo hacer nada.
- **posa = posb** pero **A[i] != B[i]**. En este caso, este es el último nodo en común entre ambos caminos. Por tanto, este nodo tendrá un hijo que pertenece al camino de **A**, uno que pertenece al camino de **B**, y puede tener otros hijos en el medio. En este caso sumaremos a la solución los valores de los hijos del nodo actual que estén en el medio entre los pertenecientes a **A** y **B**.

---

<sup>1</sup> <https://es.wikipedia.org/wiki/Trie>

# CONSTRUYENDO PUEBLOS

AUTOR: LAUTARO LASORSA

- **posa**  $\neq$  **posb**. En este caso, todos los nodos que estén a la derecha del hijo de **posa** correspondiente a **A** estarán necesariamente a la izquierda de **B**, y análogamente los que están a la izquierda del hijo de **posb** correspondiente a **B**, estarán a la derecha de **A**. Por tanto, sumaré sus valores a la respuesta (como **posa**  $\neq$  **posb**, no hay riesgo de contar un nodo 2 veces).

Si recorro el **Trie** simultáneamente en ambos caminos teniendo en cuenta los 3 casos, llegaré a la respuesta.

Como en cada caso la cantidad de operaciones es lineal en relación a **M**, la complejidad total queda **N\*M**. Lo que entra claramente en tiempo.

## Soluciones Parciales

La primer subtarea apunta a la solución de fuerza bruta. Que consiste en mirar cada uno de los caminos ya ingresados y ver si corresponde con el que se desea cambiar / está contenido en el rango que nos interesa.

La segunda subtarea puede ser respondida utilizando un map donde para camino ya ingresado se guarde el valor relativo al mismo. Como la función **Crear** siempre devolverá 0, el competidor no deberá preocuparse mucho por ella.

Para la tercera subtarea solo será necesaria la observación del caso **posa** = **posb** pero **A[i]**  $\neq$  **B[i]**, ya que al diferir en una sola posición el otro caso nunca puede darse.

En la cuarta subtarea el caso **posa** = **posb** pero **A[i]**  $\neq$  **B[i]** nunca abarcara ningún nodo intermedio, por lo cual solo será necesaria la observación del caso **posa**  $\neq$  **posb**, la cual además se ve simplificada en su implementación.