Diseño y Análisis de Algoritmos - Proyecto
PROBLEMA B

Sebastian Puerto
201318518

Diciembre 10 , 2019

1. Algoritmo de Solución

IDEA GENERAL: Se ordena la lista de personas de acuerdo a su valor de A, y luego de haya la subsecuencia maximal de menor orden lexicográfico descendiente estrictamente para la cadena de B's que resultante.

Que la la cadena resultante satisfaga ser ascendente para A y descendiente para B se debe a que cualquier subsecuencia de B's escogida luego de ordenar respecto a A será una cadena ascendente para A, así que al escogerla estrictamente descendiente para B se satisfacerán las dos condiciones simultáneamente. Que la cadena sea de longitud máxima y la menor lexicográficamente se debe a detalles del algoritmo que se discutirán a continuación.

Comentario: Inicialmente se consideró ordenar descendentemente la lista respecto a A y luego aplicar un algoritmo estándar para encontrar la subsecuencia estrictamente creciente de la lista de B's resultante, algoritmo que conserva última subsecuencia maximal encontrada, sin embargo se pudo ver con ejemplos puntuales que esto no garantiza que se obtenga aquella de menor orden lexicográfico.

El algoritmo de solución itera sobre cada persona de la lista ordenada respecto a A. Se conserva una Lista Enlazada que contiene todas las subsecuencias que tienen el potencial de convertirse en la cadena maximal de menor orden lexicográfico; la lista enlazada de subsecuencias candidatas tiene las siguientes propiedades:

- La longitud de las subsecuencias candidatas tienen un orden descreciente no estricto (respecto al índice de la lista enlazada).
- La lista compuesta de las colas (elemento final) de cada subsecuencia es una lista estrictamente descreciente.
- Entre las subsecuencia candidatas de longitud l, ninguna es lexicográficamente mayor a la subsecuencia de esta longitud que tiene el mayor índice en la lista enlazada.

Como se mencionó, el algoritmo de solución itera sobre las personas de la lista ordenada respecto a A. Al ver este nuevo elemento,

- Se busca el primer índice en la lista de candidatos en donde podría agregarse el elemento actual (encontrando la primera cola que es estrictamente mayor al valor actual de B); este índice se llamará *ii*.
- Se recorren los candidatos con la misma longitud y que tengan un índice mayor en la lista de candidatos (lo cual significa que el elemento actual de B también puede ser agregado a estas subsecuencias pues la lista de colas es estrictamente descresciente), buscando entre estos candidatos el que es lexicográficamente menor; el índice en el cual se encuentra se llamará *imL*. *Solamente vale la pena conservar este candidato* de longitud maximal entre los que pueden ser actualizados (es decir, se les puede agregar el elemento actual de B) pues será cierto que:

- Al tener un índice menor que todos aquellos , cualquier subsecuencia decresciente estricta para B que extienda a algún candidato entre ii (inclusivo) y imL (exclusivo) también extenderá al candidato imL.
- Al ser lexicográficamente menor, cualquier extensión de un candidato entre ii (inclusivo) y imL (exclusivo) será necesariamente lexicográficament menor a cualquier extensión del candidaato en imL.
- Por lo tanto, sabemos que las extensiones de los candidatos entre ii (inclusivo) y imL (exclusivo) no tienen posibilidad de ser la respuesta final ya que cualquier extensión de ellas o será más larga o será mayor lexicográficamente a la extensión del candidato en imL.

- Ahora se borran todas las subsecuencias entre ii (inclusivo) y imL (exclusivo).
- Se busca el último índice de las cadenas con longitud 1 más que el candidato que se ha escogido. Una copia de este candidato con la adición del nuevo elemento en la cola será agregado como el nuevo último elemento de los candidatos que tienen esta longitud. Al hacer esto, se conserva que la lista de candidatos tenga lista de colas estrictamente decrecriente.
- Para que también se satisfaga el invariante de no haber cadenas de la misma longitud mayores lexicográficamente al último candidato, se borran aquellas que son lexicográficamente mayores entre esta longitud.
    - Esto se puede hacer por las mismas 2 razones que se borraron las cadenas que tuvieron los índices ii (inclusivo) y imL (exclusivo).

Creo que es de hecho cierto que al realizar estas 2 operaciones se están conservando únicamente aquellas subsecuencias que pueden ser extendidas por personas de próximas iteraciones para convertirse en la secuencia de respuesta. Podría también pensarse en eliminar las subsecuencias de tamaño que, debido a cuántas personas quedan por examinar, no tengan posibilidad de ser la respuesta.

Al finalizar de recorrer la lista de personas, se busca aquella que es lexicográficamente menor entre la lista de candidatos que tienen longitud maximal.

2. Análisis de Complejidades Espacial y Temporal

El tamaño del problema está dado por N, el número de personas.

2.1. Análisis Espacial

Se hace uso una copia de la lista de personas original: $O(N)$.

Se conserva una lista enlazada con todos los posibles candidatos, la cual contiene al menos un elemento de cada longitud entre 1 y el tamaño de la subsecuencia maximal, que en el peor de los casos es N. No sabría probarlo, pero hay un número no proporcional a N de cada longitud, así que esta lista ocupa un espacio de $O(1 + 2 + \cdots + N) = O(N^2)$.

Hay variables de tamaño constante que, por lo anterior, no contribuyen al orden espacial.

En total se tiene un orden espacial de

$$S_B(N) = O(N^2)$$

Comentario: por falta de tiempo, en el código implemento la remoción de los candidatos entre ii (inclusivo) y imL (exclusivo), no los de más arriba, así que sí se están conservando más de los necesarios, y el tercer invariante no es un invariante de ese código.

2.2. Análisis Temporal

3. Comentarios Finales

Se observó que si no se eliminan los candidatos entr ii (inclusivo) y imL (exclusivo) el tiempo de ejecución puede ser hasta 7 veces mayor.