Punto 3 – Texto Mínimo Reconstruible

**Identificación:**

Juan Andrés Romero -202013449

Luccas Rojas -201923052

**Explicación:**

Para el problema del texto mínimo reconstruible se planeó en desarrollar un algoritmo greedy debido a la complejidad del problema. Se encontró que el ejercicio en cuestión puede reducirse a un problema tipo TSP (Travelling Salesman Person), donde cada subcadena es un vértice del grafo, los pesos entre nodos es el overlap o sobrelapamiento de letras entre ellas y se debe de encontrar un camino de costo máximo que pase por cada uno de los vértices una sola vez. Por lo tanto, se puede concluir que el texto mínimo reconstruible o *Shortest Superstring Problem* es un problema NP-Hard que no se puede resolver en tiempo polinómico (a menos de que P = NP, pero ese es otro debate). Con base en esto, se planeó el desarrollo de un algoritmo ávaro aproximado de decisión que no tenga una complejidad temporal tan elevada, pero que pueda llegar a dar una respuesta equivocada (que el String resultante no sea el mínimo). De esta manera se llegó a un algoritmo basado en 2 funciones auxiliares y una principal. Dentro de primeras, se encuentran overlap() e isBlank(), las cuales calculan el sobrelapamiento de dos strings y revisan si un String se encuentra vacío o no respectivamente. Overlap recibe como parámetro 2 strings cualquiera y devuelve tanto el valor del overlap (cuántas letras se sobrelapan como máximo entre los strings) y una cadena que es combinación de las entradas realizada por el overlap encontrado. En cuanto a la función principal, ésta utiliza 2 listas auxiliares, y 3 variables que representarán los resultados de cada iteración del ciclo principal. Aux, que representa la lista de strings que están siendo operadas y aux2, que representa una lista de superstrings cuyos String padre tienen el overlap máximo encontrado.

El pseudocódigo del algoritmo es el siguiente:

Copiar el arreglo de entrada en el primer arreglo auxiliar

Realizar un ciclo while mientras aux tenga una longitud diferente de 1

Encontrar el par de strings en aux que mayor overlap tengan

Reemplazar el par encontrado por la combinación de las strings

Una vez se tenga un solo String en aux, ese es el resultado.

Sin embargo, existen 2 casos en donde este procedimiento puede irse por un camino que no es. El primero ocurre cuando no existe ningún tipo de overlap entre las strings pasadas como parámetro, entonces el Superstring más pequeño es la concatenación de los substrings. Por otro lado, el segundo caso es cuando si existe un sobrelapamiento, pero las strings finales no tienen overlap entre ellas, esto quiere decir, que el algoritmo tomó el camino equivocado, y por tanto, es necesario evaluar otro. En este caso, se copia aux2 dentro de aux y se revisa si existe una String de la lista original que no haya sido fusionada en alguno de los strings de aux2. Luego se vuelve a realizar el ciclo anterior hasta dar con el resultado.

**Análisis de complejidad:**

**Comentarios:**