

Laboratorio Nro. 3

Listas Enlazadas y Vectores dinámicos

Alejandra Palacio Jaramillo
Universidad Eafit
Medellín, Colombia
apalacioj@eafit.edu.co

Valentina Moreno Ramírez
Universidad Eafit
Medellín, Colombia
vmorenor@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1 La complejidad del ejercicio es de $O(n + m)$ siendo n el número de línea a leer y el m el número de palabras, siendo lineal.

Para este ejercicio se implementó la estructura de datos denominada vector dinámico, ya que se deduce que, según el problema dado, se va a necesitar principalmente el acceso a los datos de este, por ende, teniendo en cuenta las complejidades de otras estructuras de datos como LinkedList, se tomó la decisión de que la más acertada sería esta.

3.2 El ejercicio 2.1 es solucionado por el método *tecladoRoto()* de la clase *ejercicioEnLinea*, en el cual primero se crea una lista enlazada de caracteres llamada *texto*, dos variables booleanas llamadas *inicio* y *firstTime*, una cadena de caracteres vacía llamada *texto1*, un entero *j* inicializado en cero y un objeto de la clase *StringBuilder* llamado *s1*, el cuál se crea con base en la cadena de caracteres que se le pasa como argumento, en este caso, la cadena que se quiere ordenar de acuerdo a la forma indicada por sus ']' y '['. En primera instancia, se tiene un ciclo *for* que recorre todo el *String* contenido en *s1*. Luego, se tienen dos *if*'s anidados, los cuales tiene como función eliminar a ']' o '[' si estos se encuentran en la posición indicada por el índice que señala el ciclo en el que están contenidos. Además, dentro cada *if* se crea un *StringBuilder auxiliar*, el cual tendrá a *s1* pero sin el carácter eliminado (']' o '['). Adicionalmente, dentro de cada uno de estos *if*'s se encuentra otro *if* que sirve para revisar si en la siguiente posición también hay ']' o '[' y, de ser esto verdadero, se obtiene que el *índice i* será igual a *índice i menos 1* y que *s1* será igual a *auxiliar*, todo esto para continuar con la respectiva iteración en el *s1* nuevo. Cabe destacar que en cada uno de estos *if*'s se agrega el carácter en la posición *i* (luego de la eliminación de los ']' o '[') en el inicio de la lista *texto* o al final de esta y se asignan nuevos valores a *inicio* y *firstTime*, según sea el caso. Ahora bien, si ninguna de las condiciones en los *if*'s anteriores se cumple, se evalúa en otro *if* a *inicio* y a *firstTime*. Si *inicio* es verdadero (es decir, si el punto de inserción se encuentra al inicio de la lista luego de agregar un primer carácter en la posición 0 de la misma) y si *firstTime* es falso (es decir, si ya se agregó por primera vez un carácter al inicio de la lista) entonces se agregarán los caracteres siguientes luego de este primer carácter agregado. Para esto, *firstTime* continúa siendo falso por las razones expuestas en líneas

ESTRUCTURA DE DATOS 1

Código ST0245

anteriores y se utiliza al entero j , el cuál sirve como el indicador del índice de la posición en la cual se quiere agregar al carácter que se encuentra en la posición i en ese momento. Este indicador se aumenta en una unidad cada que este *if* sucede, se inicializa en cero cada que entra al primer *if* mencionado antes (al de '[') y se aumenta también en una unidad cada que ocurre *firstTime*, es decir, cada que se agrega un carácter al inicio de la lista *texto*.

Ejecutado el código descrito en el párrafo anterior, se tiene la lista ordenada de acuerdo con las especificaciones dadas. Ahora, lo que queda es agregar cada uno de los caracteres que se encuentran en la lista al *String texto1*, lo cual se logra con un *ciclo for* que recorre la lista y obtiene cada carácter en la posición i para agregarlo a este *String*. Por último, se retorna el *String* que contiene los caracteres de la lista enlazada, en otras palabras, se retorna a *texto1*.

3.3 $T(n,m) = n^2 + m^2$

Como m puede ser menor que n (en el código *ejercicioEnLinea* hay una explicación más detallada de esta conclusión), entonces se tiene que:

$T(n,m) = O(n^2 + m^2)$, por regla de reflexividad.

$O(n^2 + m^2) = O(n^2)$, por regla de la suma (recordar que $n > m$).

La complejidad asintótica para el peor de los casos es cuadrática.

Por otro lado, m también puede ser igual a n (en el código *ejercicioEnLinea* hay una explicación más detallada de esta conclusión), entonces se tiene que:

$T(n,m) = O(n^2 + m^2)$, por regla de reflexividad.

$O(n^2 + m^2) = O(2n^2)$, por suma de variables iguales (recordar que $n=m$).

$O(2n^2) = O(n^2)$, por regla del producto

La complejidad asintótica para el peor de los casos es cuadrática.

3.4

$n \rightarrow$ longitud de la cadena de caracteres que se quiere ordenar de acuerdo a los '[' y ']' que contiene en ella.

$m \rightarrow$ longitud de la lista enlazada que contiene la cadena de caracteres luego de la eliminación de los '[' y '[' y del ordenamiento de la misma con base a lo definido por '[' y '['.

Teniendo esto en cuenta, a su vez se puede confirmar lo descrito en el numeral 3.3 y en el código en *ejercicioEnLinea.java*

4) Simulacro de Parcial

4.1.1 B

4.1.2 B

4.2 C

4.3 Según el documento se encuentra pendiente el ejercicio

4.4.1 Token

4.4.2 C

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas

Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627

Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 1
Código ST0245

4.5 A

4.6 B

4.7 *Según el documento se encuentra pendiente el ejercicio*

4.8 D

4.8.1 A

4.8.2 C

4.8.3 C

4.9.1 D

4.9.2 A

4.9.3 B

4.10.1 B

4.10.2 B

4.11.1 *!s1.isEmpty()*

4.11.2 *s1.pop()*

4.11.3 *s2.pop()*

4.12.1 IV

4.12.2 I

4.13.1 III

4.13.2 III

4.14. IV

5) Lectura recomendada (opcional)

Mapa conceptual

6) Trabajo en Equipo y Progreso Gradual (Opcional)

6.1 *Actas de reunión*

6.2 *El reporte de cambios en el código*

6.3 *El reporte de cambios del informe de laboratorio*

PhD. Mauricio Toro Bermúdez

Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473