

Ejercicio tipo parcial

Cuando se debe procesar texto expresado en lenguaje natural, una de las primeras tareas que se realiza es la de eliminación de *stopwords*. Las stopwords son palabras muy comunes en un determinado lenguaje, por ejemplo, artículos (el, la, las, los...), preposiciones (a, ante, con, por...), etc.

Dado un texto almacenado en un arreglo de caracteres de tamaño MAX_A, donde cada palabra está delimitada por espacios, comas (,) o puntos (.) se pide eliminar todas las stopwords almacenadas en otro arreglo de caracteres de tamaño MAX_S que se encuentra delimitado por espacios.

Por ejemplo, dado un texto almacenado en un arreglo A:

		I	a		c	a	s	a		r	o	j	a	,		a		l	a		v	u	e	l	t	a		d	e		l	a		e	s	q	u	i	n	a	.		
--	--	---	---	--	---	---	---	---	--	---	---	---	---	---	--	---	--	---	---	--	---	---	---	---	---	---	--	---	---	--	---	---	--	---	---	---	---	---	---	---	---	--	--

Y una lista de stopwords S

	a		l	o		l	o	s		d	e		l	a		l	a	s	
--	---	--	---	---	--	---	---	---	--	---	---	--	---	---	--	---	---	---	--

El arreglo A debería quedar de la siguiente forma luego de la eliminación de stopwords:

[illegible]

RESOLUCION

```

public class ejercicioTipoParcial {
    public static final int MAX_A = 42;
    public static final int MAX_S = 20;
    public static final char SEPARADOR1=' ';
    public static final char SEPARADOR2=',';
    public static final char SEPARADOR3='.';
    public static void main(String[] args) {
        char[] arrA = {' ','l','a',' ','c','a','s','a',' ','r','o','j','a',' ',' ','a',' ','l','a',' ','v','u','e','l','t','a',' ','d','e',' ','l','a',' ','e','s','q','u','i','n','a','.', ' '};
        char[] arrS = {' ','a',' ','l','o',' ','l','o','s',' ','d','e',' ','l','a',' ','l','a','s',' '};
        int ini = 0, fin = -1;
        while (ini < MAX_A) {
            ini = buscarInicio(arrA, fin + 1, MAX_A);
            if (ini < MAX_A) {
                fin = buscarFin(arrA, ini, MAX_A);
                if (esStopword(arrA, ini, fin, arrS)) {
                    eliminarPalabra(arrA, ini, fin);
                    fin = ini;
                }
            }
        }
        mostrarArreglo(arrA);
    }
    public static int buscarInicio(char[] arr, int pos, int MAX) {
        while (pos < MAX && (arr[pos] == SEPARADOR1 || arr[pos] == SEPARADOR2 || arr[pos] == SEPARADOR3))
            pos++;
        return pos;
    }
    public static int buscarFin(char[] arr, int pos, int MAX) {
        while (pos < MAX && (arr[pos] != SEPARADOR1 && arr[pos] != SEPARADOR2 && arr[pos] != SEPARADOR3))

```

```

        pos++;
        return pos - 1;
    }
    public static boolean esStopword(char[] arrA, int ini, int fin, char[] arrS) {
        int iniS = 0;
        int finS = -1;
        boolean laEncontre = false;
        while (iniS < MAX_S && !laEncontre) {
            iniS = buscarInicio(arrS, finS + 1, MAX_S);
            if (iniS < MAX_S) {
                finS = buscarFin(arrS, iniS, MAX_S);
                if ((fin-ini+1==finS-iniS+1) && sonIguales(arrA, ini, fin, arrS, iniS,
finS))

                    laEncontre = true;
            }
        }
        return laEncontre;
    }
    public static boolean sonIguales(char[] arrA, int ini, int fin, char[] arrS, int
iniS, int finS) {
        while (ini <= fin && arrA[ini] == arrS[iniS]) {
            ini++;
            iniS++;
        }
        return ini > fin;
    }
    public static void eliminarPalabra(char[] arrA, int ini, int fin) {
        for (int i = ini ; i <= fin; i++)
            correrAIzquierda(arrA, ini);
    }
    public static void correrAIzquierda(char[] arr, int pos) {
        for (int i = pos; i < MAX_A - 1; i++)
            arr[i] = arr[i + 1];
    }
    public static void mostrarArreglo(char[] arr) {
        for (int i = 0; i < MAX_A; i++)
            System.out.print(arr[i] + "|");
        System.out.println();
    }
}

```