



Programación 1

Extra Tipo Parcial

Enunciado


- 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:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|---|---|--|---|---|---|---|--|---|---|---|---|---|--|---|--|---|---|--|---|---|---|---|---|---|--|---|---|--|---|---|--|---|---|---|---|---|---|---|---|
| | 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 | . |
|--|---|---|--|---|---|---|---|--|---|---|---|---|---|--|---|--|---|---|--|---|---|---|---|---|---|--|---|---|--|---|---|--|---|---|---|---|---|---|---|---|

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

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

Estrategia



```
mientras no se termina de recorrer A
    busco inicio de secuencia desde la última
    si inicio no supera el máximo
        busco el fin de la secuencia
        si es stop word
            eliminar Palabra

mostrarArreglo
```

```
public static void main(String[] args) {
    char[] arrA = ...;
    char[] arrS = ...;
    int ini = 0;
    int fin = -1;

    while (ini < MAX_A) {
        ini = buscarInicioA(arrA, fin + 1);
        if (ini < MAX_A) {
            fin = buscarFinA(arrA, ini);
            if (esStopword(arrA, ini, fin, arrS)) {
                eliminarPalabra(arrA, ini, fin);
                fin = ini;
            }
        }
    }
    mostrarArreglo(arrA);
}
```

Código

```
public static final int MAX_A = 41;
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 = {...};
    char[] arrS = {...};
    int ini = 0;
    int fin = -1;

    while (ini < MAX_A) {
        ini = buscarInicioA(arrA, fin + 1);
        if (ini < MAX_A) {
            fin = buscarFinA(arrA, ini);
            if (esStopword(arrA, ini, fin, arrS)) {
                eliminarPalabra(arrA, ini, fin);
                fin = ini;
            }
        }
    }
    mostrarArreglo(arrA);
}
```

Código - Métodos y Funciones conocidos



```
private static int buscarInicioA(char[] arr, int pos) {
    while (pos < MAX_A && (arr[pos] == SEPARADOR1 || arr[pos] == SEPARADOR2 || arr[pos] == SEPARADOR3))
        pos++;
    return pos;
}

private static int buscarFinA(char[] arr, int pos) {
    while (pos < MAX_A && (arr[pos] != SEPARADOR1 && arr[pos] != SEPARADOR2 && arr[pos] != SEPARADOR3))
        pos++;
    return pos - 1;
}

private static int buscarIniS(char[] arr, int pos) {
    while (pos < MAX_S && arr[pos] == SEPARADOR1)
        pos++;
    return pos;
}

private static int buscarFinS(char[] arr, int pos) {
    while (pos < MAX_S && arr[pos] != SEPARADOR1)
        pos++;
    return pos - 1;
}
```

Código - Métodos y Funciones conocidos



```
private static void eliminarPalabra(char[] arrA, int ini, int fin) {  
    for (int i = ini ; i <= fin; i++)  
        correrAlzquierda(arrA, ini);  
}
```

```
private static void correrAlzquierda(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();  
}
```

Código



```
private 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 = buscarIniS(arrS, finS + 1);
        if (iniS < MAX_S) {
            finS = buscarFinS(arrS, iniS);
            if (sonIguales(arrA, ini, fin, arrS, iniS, finS))
                laEncontre = true;
        }
    }
    return laEncontre;
}
```

Código



```
private static boolean sonIguales(char[] arrA, int ini, int fin, char[] arrS, int iniS, int finS) {
    if (fin - ini + 1 == finS - iniS + 1) {
        while (ini <= fin && arrA[ini] == arrS[iniS]) {
            ini++;
            iniS++;
        }
        return ini > fin;
    }
    else
        return false;
}
```



```

public class TipoParcial1 {
    public static final int MAX A = 41;
    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;
        int fin = -1;

        while (ini < MAX A) {
            ini = buscarInicioA(arrA, fin + 1);
            if (ini < MAX A) {
                fin = buscarFinA(arrA, ini);
                if (esStopword(arrA, ini, fin, arrS)) {
                    eliminarPalabra(arrA, ini, fin);
                    fin = ini;
                }
            }
        }
        mostrarArreglo(arrA);
    }

    private static int buscarInicioA(char[] arr, int pos) {
        while (pos < MAX_A && (arr[pos] == SEPARADOR1 || arr[pos] == SEPARADOR2 || arr[pos] == SEPARADOR3))
            pos++;
        return pos;
    }

    private static int buscarFinA(char[] arr, int pos) {
        while (pos < MAX_A && (arr[pos] != SEPARADOR1 && arr[pos] != SEPARADOR2 && arr[pos] != SEPARADOR3))
            pos++;
        return pos - 1;
    }
}

```

```

private 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 = buscarIniS(arrS, finS + 1);
        if (iniS < MAX_S) {
            finS = buscarFinS(arrS, iniS);
            if (sonIguales(arrA, ini, fin, arrS, iniS, finS))
                laEncontre = true;
        }
    }
    return laEncontre;
}

private static boolean sonIguales(char[] arrA, int ini, int fin, char[] arrS, int iniS, int finS) {
    if (fin - ini + 1 == finS - iniS + 1) {
        while (ini <= fin && arrA[ini] == arrS[iniS]) {
            ini++;
            iniS++;
        }
        return ini > fin;
    }
    else
        return false;
}

private static int buscarIniS(char[] arr, int pos) {
    while (pos < MAX_S && arr[pos] == SEPARADOR1)
        pos++;
    return pos;
}

private static int buscarFinS(char[] arr, int pos) {
    while (pos < MAX_S && arr[pos] != SEPARADOR1)
        pos++;
    return pos - 1;
}

```

```
private static void eliminarPalabra(char[] arrA, int ini, int fin) {  
    for (int i = ini ; i <= fin; i++)  
        correrAlzquierda(arrA, ini);  
}  
  
private static void correrAlzquierda(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();  
}  
  
}
```