Estructura de Datos y Algoritmos

ITBA 2024-Q2

TP 2-Ejer 2.1, 2.2, 2.3 y 2.4

Implementar Soundex

Ejemplos

Soundex ("threshold").representation() //"T624"

Soundex ("hold").representation() //"H430"

Soundex ("phone").representation() //"P500"

Soundex ("foun").representation() //"F500"

Hagamos casos de testeo!

```
@Test
void soundexRepresentation_threshold_Test() {
    assertEquals("T624",Soundex.representation("threshold"));
}
@Test
void soundexRepresentation_hold_Test() {
    assertEquals("H430",Soundex.representation("hold"));
}
@Test
void soundexRepresentation_phone_Test() {
    assertEquals("P500",Soundex.representation("phone"));
}
@Test
void soundexRepresentation_foun_Test() {
    assertEquals("F500",Soundex.representation("foun"));
}
```

Implementemos el algoritmo

Repasemos entre todos el algoritmo.

```
public static String representation(String s){
    s = s.toUpperCase();
    char[] IN = s.toCharArray();
    char[] OUT = {'0','0','0','0'};
    OUT[0] = IN[0];
    int count = 1;
    char current, last = getMapping(IN[0]);
    for(int i=1; i < IN.length && count < 4; i++){
        char iter=IN[i];
        current = getMapping(iter);
        if(current != '0' && current !=last)
            OUT[count++] = current;
        last = current;
    }
    return new String(OUT);
}
```

¿ Qué opciones se les ocurren para getMapping ?

Implementemos el algoritmo

Ahora implementen ustedes la representación y la similitud (ej 2.1, 2.2, 2.3, y 2.4)

Hay mejores fonéticos?

Metaphone

Buscar en Wikipedia English:

"**Metaphone**" y explicar algunas de las mejoras más relevantes que introduce.

Metaphone

Importante de Metaphone:

El encoding genera símbolos de **longitud arbitraria**.

Cómo usarlo?

Ej: Metaphone (“threshold”) = 0RXLT

Metaphone(“hold”) = HLT

Metaphone(“zresjoulding”) = SRSJLTNK

Metaphone(“phone”) = FN

Metaphone (“foun”) = FN

SimilitudMetaphone (“phone”, “fown”) = 1 (mejoró!!!)

Metaphone

Cómo medimos la similitud?

La similitud entre 2 textos puede pensarse como la proporción de caracteres coincidentes entre los encodings respecto de la máxima longitud de los encodings obtenidos, ya que son de longitud variable. Pero hay algo mejor… en un rato lo discutimos.