

Estructura de Datos y Algoritmos

ITBA 2024-Q2

TP 2B- Ejer 3.1



En la clase KPM implementar el método de clase `indexOf` que aprovecha la tabla de Next (algoritmo Knuth-Morris-Pratt)

```

public static int indexOf( char[] query, char[] target) {
    if (query == null || query.length== 0)
        throw new RuntimeException("Bad query string");
    if (target == null | target.length == 0)
        throw new RuntimeException("Bad target string");

    int[] next = nextComputation(query);

    int rec= 0;
    int pquery= 0;

    while ( rec < target.length ) {
        if (target[rec]==query[pquery] ) {
            rec++;
            pquery++;
        }

        if ( pquery == query.length) { // Found!!!
            break;
        }
        else // mismatch?
            if ( rec < target.length && target[rec]!=query[pquery])
                // no machea los i-1
                {
                    if ( pquery != 0)
                        pquery= next[ pquery - 1];
                    else
                        rec++;
                }
    }

    if ( pquery == query.length)
        return rec - pquery;
    else
        return -1;
}

```

TP 2B- Ejer 4.1 y 4.2



En la clase KPM implementar el método de clase

```
ArrayList<Integer>  
findAll(char[] query, char[]  
target)
```

que aprovecha la tabla de Next (algoritmo Knuth-MorrisPratt) para calcular todas las posiciones donde ocurre query dentro de target

Escribir testeos correspondiente!!!

Caso de Uso:

KPM.findAll("no", "sino se los digo no se si es nocivo") debería devolver una instancia de arraylist con las posiciones 2, 17, 29

KPM.findAll("ni", "sino se los digo no se si es nocivo") debería devolver un empty arraylist

¿Probaron qué pasa cuando se solapa?

Si no lo consideraron, agregar un testeo para

Caso de Uso

```
System.out.println( KPM.findAll("aaa".toCharArray(),  
"aaabaaaaab".toCharArray()) );
```

```
// aunque se solapen debe devolver
```

```
// [0, 4, 5, 6]
```