private int[] right;

private String pat;

public BoyerMoore(String pat) {

this.pat = pat;

int M = pat.length();

int R = 256; // The radix

// Initialize the right array to store the rightmost position of each character in the pattern

right = new int[R];

for (int c = 0; c < R; c++) {

right[c] = -1; // -1 means the character is not in the pattern

}

for (int j = 0; j < M; j++) {

right[pat.charAt(j)] = j; // The rightmost position in the pattern

}

}

public int search(String txt) {

int N = txt.length();

int M = pat.length();

int skip;

for (int i = 0; i <= N - M; i += skip) {

skip = 0;

for (int j = M - 1; j >= 0; j--) {

if (pat.charAt(j) != txt.charAt(i + j)) {

skip = Math.max(1, j - right[txt.charAt(i + j)]);

break;

}

}

if (skip == 0) return i; // Pattern found at position i

}

return N; // Pattern not found

}

public static void main(String[] args) {

String txt = "HERE IS A SIMPLE EXAMPLE";

String pat = "EXAMPLE";

BoyerMoore bm = new BoyerMoore(pat);

int position = bm.search(txt);

if (position < txt.length()) {

System.out.println("Pattern found at index " + position);

} else {

System.out.println("Pattern not found");

}

}

}