

33|

Word

BF

RK

BF

RK

BF

RK

BM Boyer-Moore

KMP 3 4 BM

BM

BF

RK

c

c

c

BM

BM

BM

bad character rule

good suffix shift

1.

c

c

c

a

0

d

a

a

a

xi -1

si-xi

si

xi

2.

BM

BM

 $O(n/m)$

xi

aaabaaabaaabaaab

aaaa

aaaaaaaaaaaaaaaa

baaa

BM

“ Si-Xi ”

2

3

{u}

bc

{u}

{u}

{u*}

{u*}

{u}

{u}

{u}

{u}

{u*}

{u}

bc

{u}

{u}

a ab s

s

abc

c, bc

{v}

s

abc

BM

BM

“ ”

si-xi xi xi

ASCII

1

256

b

m

bc

```
private static final int SIZE = 256; //
private void generateBC(char[] b, int m, int[] bc) {
    for (int i = 0; i < SIZE; ++i) {
        bc[i] = -1; // bc
    }
    for (int i = 0; i < m; ++i) {
        int ascii = (int)b[i]; // b[i] ASCII
```



```
    bc[ascii] = i;
  }
}
```

BM

si-xi

```
public int bm(char[] a, int n, char[] b, int m) {
  int[] bc = new int[SIZE]; //
  generateBC(b, m, bc); //
  int i = 0; // i
  while (i <= n - m) {
    int j;
    for (j = m - 1; j >= 0; --j) { //
      if (a[i+j] != b[j]) break; //
    }
    if (j < 0) {
      return i; //
    }
    //
    i = i + (j - bc[(int)a[i+j]]);
  }
  return -1;
}
```

•

•

“ ”

BM

m-1

suffix

suffix

k

{u}

{u*}

1

{u}

suffix

boolean

suffix
prefix

suffix

0 i i 0 m-2
j 0

prefix[k]=true k suffix[k]=j j

suffix prefix

```
// b                    m                    suffix    prefix
private void generateGS(char[] b, int m, int[] suffix, boolean[] prefix) {
    for (int i = 0; i < m; ++i) { //
        suffix[i] = -1;
        prefix[i] = false;
    }
    for (int i = 0; i < m - 1; ++i) { // b[0, i]
        int j = i;
        int k = 0; //
        while (j >= 0 && b[j] == b[m-1-k]) { //    b[0, m-1]
            --j;
            ++k;
            suffix[k] = j+1; //j+1                    b[0, i]
```

'' p

```
    }  
    i  
    if (j == -1) prefix[k] = true; //  
  }  
}
```

suffix[k]+1 j k suffix suffix[k] -1 -1 j-

r b[r, m-1] r j+2 m-1 k=m-r prefix[k] true k

m

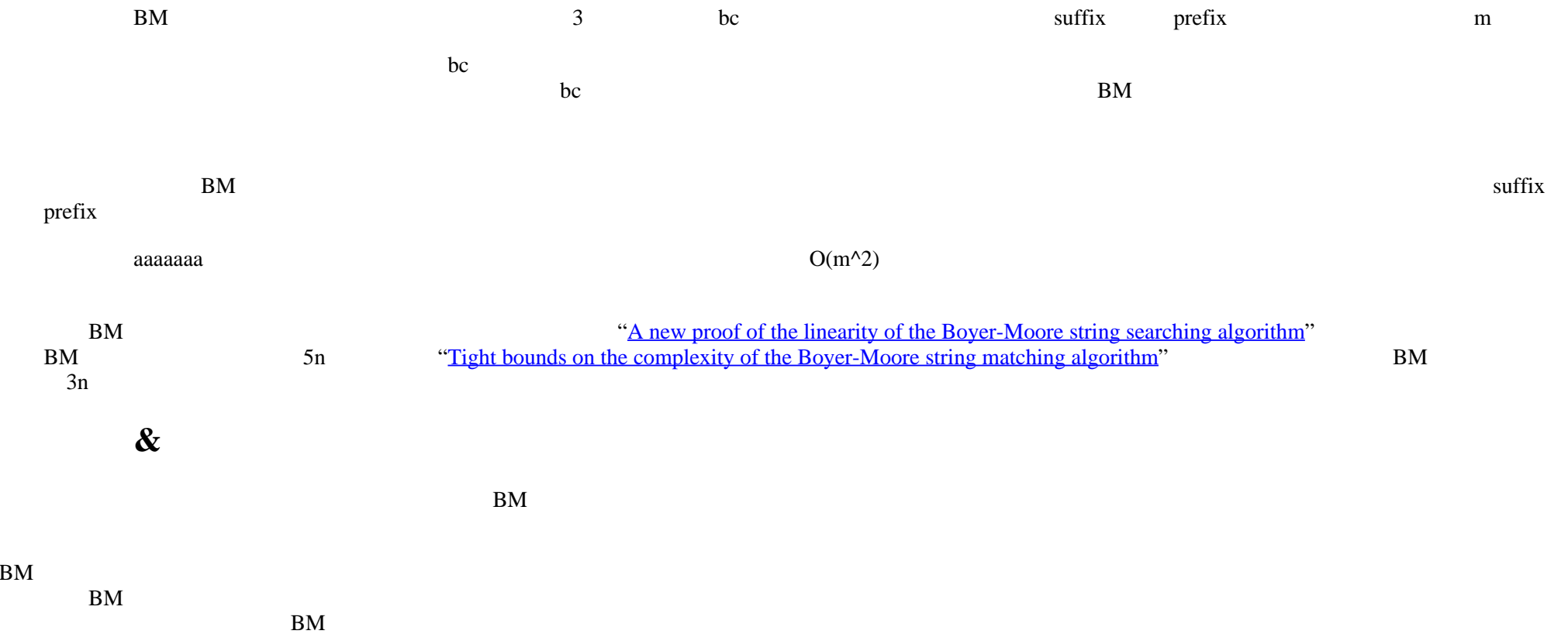
BM

```
// a,b          n    m
public int bm(char[] a, int n, char[] b, int m) {
    int[] bc = new int[SIZE]; //
    generateBC(b, m, bc); //
    int[] suffix = new int[m];
    boolean[] prefix = new boolean[m];
    generateGS(b, m, suffix, prefix);
    int i = 0; // j
    while (i <= n - m) {
        int j;
        for (j = m - 1; j >= 0; --j) { //
            if (a[i+j] != b[j]) break; //
        }
        if (j < 0) {
            return i; //
        }
        int x = j - bc[(int)a[i+j]];
        int y = 0;
        if (j < m-1) { //
```

```
        y = moveByGS(j, m, suffix, prefix);
    }
    i = i + Math.max(x, y);
}
return -1;
}

// j
private int moveByGS(int j, int m, int[] suffix, boolean[] prefix) {
    int k = m - 1 - j; //
    if (suffix[k] != -1) return j - suffix[k] + 1;
    for (int r = j+2; r <= m-1; ++r) {
        if (prefix[m-r] == true) {
            return r;
        }
    }
    return m;
}
```

BM



“[A new proof of the linearity of the Boyer-Moore string searching algorithm](#)”

“[Tight bounds on the complexity of the Boyer-Moore string matching algorithm](#)”

BM

“ ”

- Smallfly 2018-12-08 09:48:37

BM

1 BF RK BM

2

3

Get [24]

2018-12-10 02:00:29

- meng 2018-12-09 02:48:32 http://www.cs.jhu.edu/~langmea/resources/lecture_notes/boyer_moores.pdf [18]

- Liam 2018-12-07 04:51:58 $m+1, j+1$ [7]

2018-12-07 06:39:10

- meng 2018-12-23 04:16:41 $prefix[k]==0$ [6]

- Jerry 2018-12-07 05:02:53

'' p

1. ()

2. [6]

● 2018-12-10 10:06:39

BM

| | | | | | | | | |
|----|----|-------|----|----|----|----|----|----|
| xi | -1 | si-xi | PS | xi | si | si | si | si |
|----|----|-------|----|----|----|----|----|----|

u' u' u u' u u

1.

2.

3.

| | | | | |
|---|-------|--------|-----------|-----|
| | | suffix | k | |
| i | 0~m-2 | j | suffix[k] | j 0 |

| | | | | | | |
|--------|-----------|------|-------|---|---|---|
| prefix | prefix[k] | true | false | k | 0 | m |
|--------|-----------|------|-------|---|---|---|

[4]

- 2018-12-06 23:45:30

BM

Boyer Moore

n ()

$$m(\quad)$$

- suke 2018-12-19 06:35:29

a b c bc

[3]

- cygnus 2018-12-08 08:37:52

generateGS suffix prefix while k

b[r, m-1] r j+2

[3]

2018-12-10 02:06:17

j

j+1

- seniusen 2018-12-07 11:34:30

m

[3]

2018-12-10 02:10:22