

1 KMP Algorithm の訂正

```
91  /* Searching */
92  i = j = 0;
X 93  while (j+m-i <= n) {
X 94      while (i > -1 && (c_cmp++ + 1) && x[i] != y[j]) {
95          printf(" mismatch: y[%d]=%c, x[%d]=%c\n", j, y[j], i, x[i]);
96          trial++;
97          printf("%d attempt, %d char count\n\n", trial, c_cmp-c_old);
98          c_old = c_cmp;
99          i = kmpNext[i];
```

2 Boyer-Moore Algorithm (入力待ちアポート機能つき)

```
1  #include <math.h>
2  #include <stdio.h>
3  #define XSIZE 1000
4  #define ASIZE 128
5  #define N 1000
6  #define M 100
7  #define MAX(a,b) ((a)> (b)) ? (a) : (b)
8
9  void BM(char *x, int m, char *y, int n);
10 void preBmBc(char *x, int m, int bmBc[]);
11 void preBmGs(char *x, int m, int bmGs[]);
12 void suffixes(char *x, int m, int *suff);
13
14 main()
15 {
16     char x[N], y[N];
17     int m, n;
18
19     printf("Please input text: ");
20     scanf("%s", &y);
21     n = strlen(y);
22
23     printf("Please input pattern: ");
24     scanf("%s", &x);
25     m = strlen(x);
26
27     printf("\n %s, %d, %s, %d\n", y, n, x, m);
28
29     BM(x, m, y, n);
30
31 }
32
33 void BM(char *x, int m, char *y, int n) {
34     int i, j, bmGs[XSIZE], bmBc[ASIZE];
35     int c_cmp, trial, c_old;
36
37     /* preprocessing */
38     preBmGs(x, m, bmGs);
39     preBmBc(x, m, bmBc);
40
41     c_cmp = 0;
42     trial = 0;
43
44     /* Searching */
45     j = 0;
46     while (j <= n - m) {
47         for (i = m - 1; i >= 0 && (c_cmp++ + 1) && x[i] == y[i + j]; --i);
48         trial++;
49         if (i < 0) {
50             printf(" Matched at y[%d]=%c and x[%d]=%c!!!\n%d attempt, %d char comparisons\n\n",
51                 j-1, y[j-i], i-1, x[i-1], trial, c_cmp-c_old);
52             j += bmGs[0];
53         }
54         else {
55             printf(" mismatch: y[%d]=%c, x[%d]=%c\n", i+j, y[i+j], i, x[i]);
56             printf("%d attempt, %d char comparisons\n\n", trial, c_cmp-c_old);
57             j += MAX(bmGs[i], bmBc[y[i + j]] - m + 1 + i);
58         }
59         c_old = c_cmp;
60     }
61     printf("\n%d attempts, %d text character comparisons\n", trial, c_cmp);
62 }
63
```

```

64 void preBmBc(char *x, int m, int bmBc[]){
65     int i;
66
67     for (i = 0; i < ASIZE; ++i)
68         bmBc[i] = m;
69     for (i = 0; i < m - 1; ++i)
70         bmBc[x[i]] = m - i - 1;
71 }
72
73 void preBmGs(char *x, int m, int bmGs[]) {
74     int i, j, suff[XSIZE];
75
76     suffixes(x, m, suff);
77
78     for (i = 0; i < m; ++i)
79         bmGs[i] = m;
80     j = 0;
81     for (i = m - 1; i >= -1; --i)
82         if (i == -1 || suff[i] == i + 1)
83             for (; j < m - 1 - i; ++j)
84                 if (bmGs[j] == m)
85                     bmGs[j] = m - 1 - i;
86     for (i = 0; i <= m - 2; ++i)
87         bmGs[m - 1 - suff[i]] = m - 1 - i;
88 }
89
90 void suffixes(char *x, int m, int *suff) {
91     int f, g, i;
92
93     suff[m - 1] = m;
94     g = m - 1;
95     for (i = m - 2; i >= 0; --i) {
96         if (i > g && suff[i + m - 1 - f] < i - g)
97             suff[i] = suff[i + m - 1 - f];
98         else {
99             if (i < g)
100                 g = i;
101                 f = i;
102                 while (g >= 0 && x[g] == x[g + m - 1 - f])
103                     --g;
104                 suff[i] = f - g;
105         }
106     }
107 }
108
109

```

3 宿題

1 前回の宿題で作成した、テキストが 20 文字以上、パターンが 8 文字以上の 10 題の問題につ

いて、各々 bmBc 表, bmGs 表, を求め、試行回数、文字の比較回数を示せ。

2 次のパターン を受理する決定性オートマトンの状態遷移図を書け

a^*ba

$a^*b \left(a \mid b \right)^* a$

感想・意見・要望 も書いて下さい。(諸君は学ぶ権利があり、講師は教える義務があります。)