「データ構造」 第 6 回 資料 & 宿題 (*Jul. 2, 2004,* 奥乃)

1 Brute-Force Algorithm

```
 1 void BF(char *x, int m, char *y, int n) {
 2     int i, j;
 3
 4     /* searching */
 5     for (j = 0; j <= n - m; ++j) {
 6        for (i = 0; i < m && x[i] == y[i + j]; ++i);
 7        if (i >= m)
 8           OUTPUT(j);
 9     }
10 }
```

2 Knuth-Morris-Pratt Algorithm (入力待ちアボート機能つき)

```
 1 #include <stdio.h>
 2 #include<fcntl.h>
 3 #include<sys/time.h>
 4 #include<sys/types.h>
 5 #include<unistd.h>
 6 #define XSIZE 1000
 7 #define N 1000
 8 #define M 100
 9 #define A 128
10
11 void KMP(char *x, int m, char *y, int n);
12 void OUTPUT(int i);
13 void preKMP(char *x, int m, int kmpNext[]);
14
15 int isready(int fd, int timeout)
16 {
17   int rc;
18   fd_set fds;
19   struct timeval tv;
20
21   FD_ZERO(&fds);
22   FD_SET(fd,&fds);
23   tv.tv_sec = timeout;
24   tv.tv_usec = 0;
25
26   rc = select(fd+1, &fds, NULL, NULL, &tv);
27   if (rc < 0)
28     return -1;
29
30   return FD_ISSET(fd,&fds) ? 1 : 0;
31 }
32
33 main()
34 {
35   char x[N], y[N];
36   char text1[100] = "A STRING SEARCHING EXAMPLE CONSISTING OF SIMPLE TEXT",
37        patn1[50] = "STRING";
38   char text2[100] = "GCATCGCAGAGAGTATACAGTACG",
39        patn2[50] = "GCAGAGAG";
40   int m, n;
41   int fd, s;
42
43   fd = STDIN_FILENO;
44
45   printf("\nPlease input text: ");
46
47   /* scanf("%s", &y) ; */
48   s = isready(fd, 5);  /* stdin の fileno */
49   if (s < 0) {
50      exit(1); /* EOF etc ... */
51   } else if (0 < s) {  /* normal input reday */
52      scanf("%s", &y);
53   } else /* s == 0 */ {
54      strcpy(y, text2);
55      printf("%s", y);
56   }
57
58   n = strlen(y);
59
```

```
 60   printf("\nPlease input pattern: ");
 61
 62   /* scanf("%s", &x) ; */
 63   s = isready(fd, 5);  /* stdin の fileno */
 64   if (s < 0) {
 65      exit(1); /* EOF etc ... */
 66   } else if (0 < s) {  /* normal input reday */
 67      scanf("%s", &x);
 68   } else /* s == 0 */ {
 69      strcpy(x, patn2);
 70      printf("%s", x);
 71   }
 72
 73   m = strlen(x);
 74
 75   printf("\n\nText is %s, %d, and pattern is %s, %d\n\n", y, n, x, m);
 76
 77   KMP(x, m, y, n);
 78
 79 }
 80
 81 void KMP(char *x, int m, char *y, int n) {
 82   int i, j, kmpNext[1000];
 83   int c_cmp, trial, c_old;
 84
 85   /* preprocessing */
 86   preKMP(x, m, kmpNext);
 87
 88   c_cmp = 0;
 89   trial = 0;
 90
 91   /* Searching */
 92   i = j = 0;
 93   while (j+m-i <= n) {
 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];
100     }
101     i++;
102     j++;
103     if (i >= m) {
104       trial++;
105       if (x[i-1] == y[j-1])
106          printf("成功 at y[%d]=%c, x[%d]=%c!\n%d 試行, %d 文字比較\n", j-1, y[j-i], i-1, x[i-1], trial, c_cmp-c_old
107       else
108     printf("    mismatch: y[%d]=%c, x[%d]=%c\n", j-1, y[j-1], i-1, x[i-1]);
109       c_old = c_cmp;
110       i = kmpNext[i];
111     }
112   }
113  printf("\n%d attempts, %d text character comparisons\n", trial, c_cmp);
114 }
115
116
117 void OUTPUT(int i) {
118   printf("%d\n", i);
119 }
120
121
122 void preKMP(char *x, int m, int kmpNext[]) {
123   int i, j;
124
125   i = 0;
126   j = kmpNext[0] = -1;
127   while (i < m) {
128     while (j > -1 && x[i] != x[j])
129       j = kmpNext[j];
130     i++;
131     j++;
132     if (x[i] == x[j])
133       kmpNext[i] = kmpNext[j];
134     else
135       kmpNext[i] = j;
136   }
137  printf("\nkmpNext Table\n\n ");
138  for(i=0;  i<=m; i++) printf("%d: %c %d, ", i, x[i], kmpNext[i]);
139  printf("\n\n");
140 }
```

3　宿題

テキストが 20 文字以上、パタンが 8 文字以上の問題を 10 個作り、各問題について、 kmpNext と試行回数、文字の比較回数を示せ。

なお、使用した 10 個の例題は次回にも使うので、記録しておくこと。