

問題 4

文字列探索とはテキスト文字列の中から、与えられたパターン文字列に一致する部分を見つけることである。文字列探索に関する以下の各問いに答えよ。ここでテキスト文字列とパターン文字列の長さをそれぞれ n , m ($m > 0$, $n > 0$) とする。また m は n に比べて十分小さいものとする。

- (1) 最も単純な文字列探索アルゴリズムは、図 1 に示すように、テキスト文字列にパターン文字列を重ねて一致するかどうかを比較し、一致しなければ重ねる位置を 1 文字ずつ移動させていく方法である。このアルゴリズムを用いた場合、最大で何回文字比較を行う必要があるか n , m を用いて表せ。またこのアルゴリズムの時間計算量（オーダー）を述べよ。

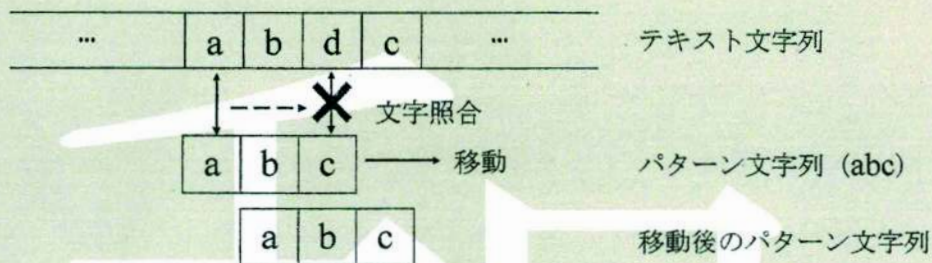


図 1

- (2) 文字の不一致が見つかったときに、パターン文字列を複数文字分移動させることによって処理を効率化できる。図 2 のように、いくつかの文字が一致した後に不一致となった場合に、その位置より前のパターンに応じて移動量が決定される。文字比較は、テキスト文字列の不一致となった文字から再開する。

図 2 の 1 回目の移動の際に 1 文字分ではなく 2 文字分移動できる理由を述べよ。また 2 回目、5 回目の移動の際に、4 文字分、5 文字分ではなく 3 文字分しか移動できない理由を述べよ。

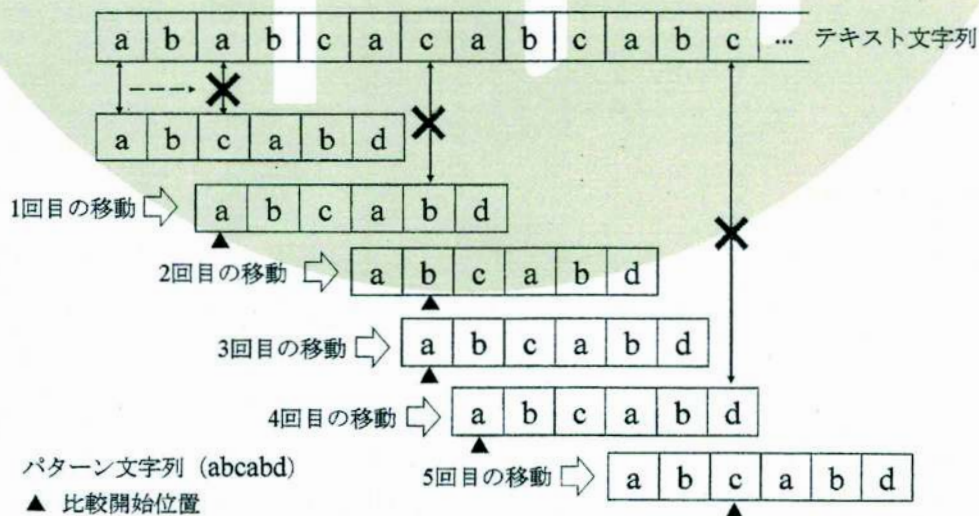


図 2

- (3) 図3に示す表はパターン文字列“abcabd”の、各文字で不一致となった場合のパターン文字列の移動量を示している。この表に倣ってパターン文字列(i)“abababcd”, (ii)“aaabcbcd”について移動量の表を作成せよ。

文字位置	0	1	2	3	4	5
パターン文字列 (abcabd)	a	b	c	a	b	d
パターン文字列移動量	1	1	2	3	3	3

図 3

- (4) プログラム1は、パターン文字列“patn”を入力として、問(3)で示した移動量の表“table”を求めるプログラムである。空欄(A), (B)に当てはまる条件を述べよ。またこのプログラムの時間計算量(オーダー)を述べよ。

プログラム1

```
void maketable(char *patn, int *table){
    int i, j, k;
    int m = strlen(patn); // パターン文字列の長さ
    table[0] = 1; table[1] = 1;
    for(i = 2; i < m; i++){
        for(j = 1; j < i; j++){
            for(k = j; k < i; k++){
                if( (A) ) break;
            }
            if( (B) ) break;
        }
        table[i] = j;
    }
}
```

- (5) 移動量の表が求まっているとして、問(2)のアルゴリズムでは文字列探索の際に最大何回文字比較を行う必要があるか述べよ。またこのアルゴリズムの時間計算量(オーダー)を述べよ。
- (6) 問(2)のアルゴリズム以外に文字列探索を高速化するアルゴリズムが数多くある。それらのアルゴリズムの中の一つを簡潔に述べよ。

Problem 4

String search is to find an occurrence of a given pattern string in a text string. Answer the following questions about string search. Let n and m ($m > 0, n > 0$) be the lengths of the text string and the pattern string, respectively, and m be much smaller than n .

- (1) One of the simplest string search algorithms compares the characters of the pattern string with those of the text string one by one as shown in Fig. 1. When a mismatch is found, the pattern string is shifted by one character. In this algorithm, find at most how many times the characters are compared using m and n . In addition, give the order of the time complexity of the algorithm.

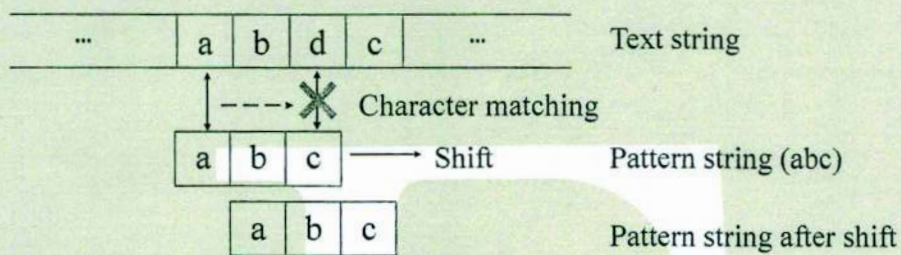


Fig. 1

- (2) The matching process can be accelerated by shifting the pattern string by more than one character when a mismatch is found. As shown in Fig. 2, when a mismatch is found after matching some characters, the shift amount is determined by the sub-pattern preceding the mismatched character. The character comparison is restarted from the mismatched character of the text string.

Describe the reason why the pattern string can be shifted by 2 characters instead of 1 character at the 1st shift in Fig. 2. In addition, describe the reason why the pattern string can be shifted by only 3 characters, by neither 4 characters nor 5 characters, when the mismatches are found in the 2nd shift and the 5th shift.

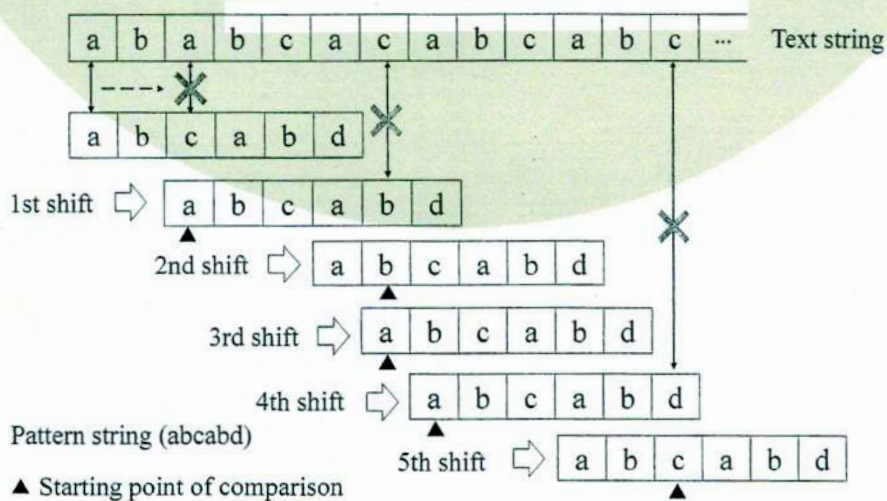


Fig. 2

- (3) Figure 3 shows the shift amounts when the mismatches are found at respective characters of the pattern string "abcabd". Make the tables of shift amounts for the pattern strings (i) "abababcd" and (ii) "aaabcabd" after the fashion of Fig. 3.

Character position	0	1	2	3	4	5
Pattern string (abcabd)	a	b	c	a	b	d
Shift amount of pattern string	1	1	2	3	3	3

Fig. 3

- (4) Program 1 is a program to make a table "table" of shift amounts for an input pattern string "patn" as explained in Question (3). Write conditions appropriate for the blanks (A) and (B). In addition, give the order of the time complexity of the program.

```

Program 1
void maketable(char *patn, int *table){
    int    i, j, k;
    int    m = strlen(patn); //    length of pattern string
    table[0] = 1; table[1] = 1;
    for(i = 2; i < m; i++){
        for(j = 1; j < i; j++){
            for(k = j; k < i; k++){
                if( [ ] (A) ) break;
            }
            if( [ ] (B) ) break;
        }
        table[i] = j;
    }
}

```

- (5) Assuming that the table of shift amounts has been made, estimate at most how many times the characters are compared by the algorithm described in Question (2). In addition, give the order of the time complexity of the algorithm.
- (6) There are many algorithms other than the algorithm described in Question (2) for accelerating string search. Describe briefly one of such algorithms.