

アルゴリズム論 (第12回)



岩手県立大学
Iwate Prefectural University

佐々木研(情報システム構築学講座)

講師 山田敬三

k-yamada@iwate-pu.ac.jp

文字列照合



岩手県立大学
Iwate Prefectural University

文字列照合問題

- 入力: テキスト S , 語 W
 - 処理: S 中に W が出現するか否かを調べる
 - 出力: 出現すればyes, そうでなければno
-
- 出力: 出現すれば出現位置, そうでなければno

2. 1 文字列の走査

	0	1	2	3	4	5	6	7	8
S	a	l	g	o	r	i	t	h	m

- 文字列は文字の配列によって表される.
- 配列の要素は0から始まる添え字によって参照される.
 - 例) $S[0]=a$, $S[2]=g$
- 文字列Sの文字数を $|S|$ と表す.
- 検索対象となる文字列を, 特にテキストという.

例題 2. 1

- テキストSがalgorithmで、文字xがtのとき、s中でxが最初に現れる位置の添字は何か.
- 答え: 6

	0	1	2	3	4	5	6	7	8
S	a	l	g	o	r	i	t	h	m

例題 2. 2

- 入力として、テキスト s と文字 x が与えられたとき、 s 中で x が最初に現れる位置の添字を求めるアルゴリズムを示せ。ただし、 x が s 中に現れないときは $|s|$ を返すこととする。

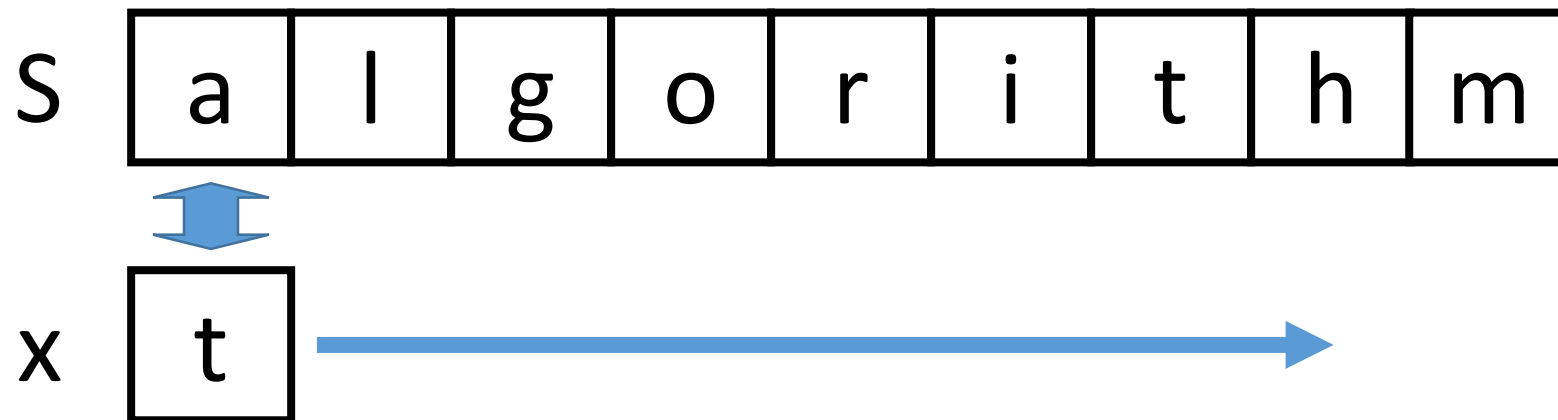
解答

Input: テキストS, 文字x

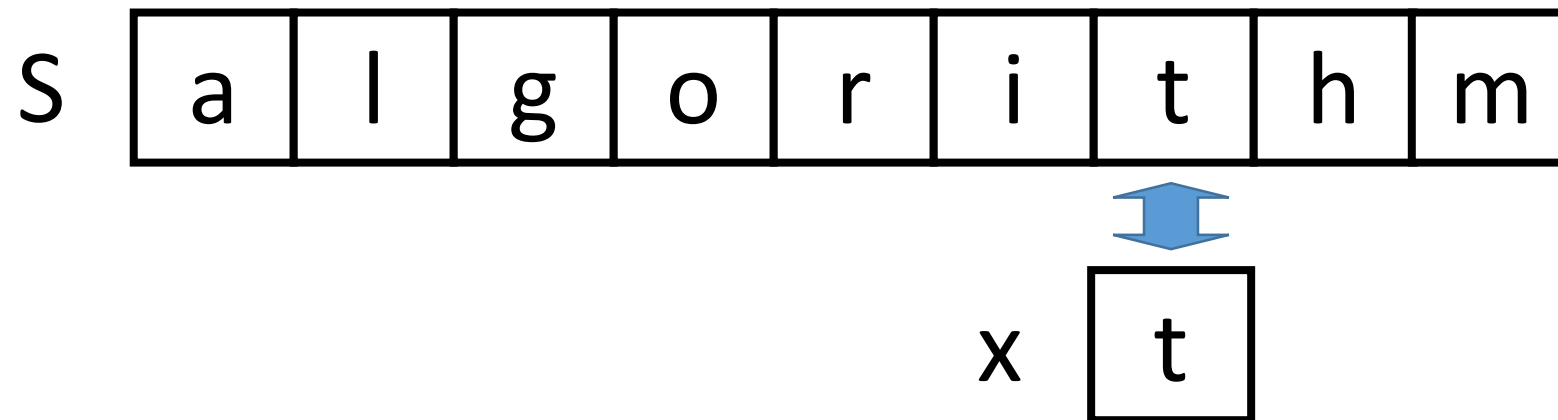
Output: テキストSにおける文字xの最初の出現位置

1. $i \leftarrow 0$
2. while $i < |S|$ do
3. if $x = S[i]$ then
4. *i* を出力して終了する
5. else
6. $i \leftarrow i + 1$
7. /* end of while */

1回目



7回目



2. 2 文字列照合問題

- 入力: テキスト s , 文字列 w
- 処理: s 中に w が出現するか否かを調べる
- 出力: 出現すれば yes, そうでなければ no
- 出力: 出現すれば出現位置, そうでなければ $|s|$
 - s が algorithm, w が go のとき, 2 を返す.

※ w を **単語** or 語 という.

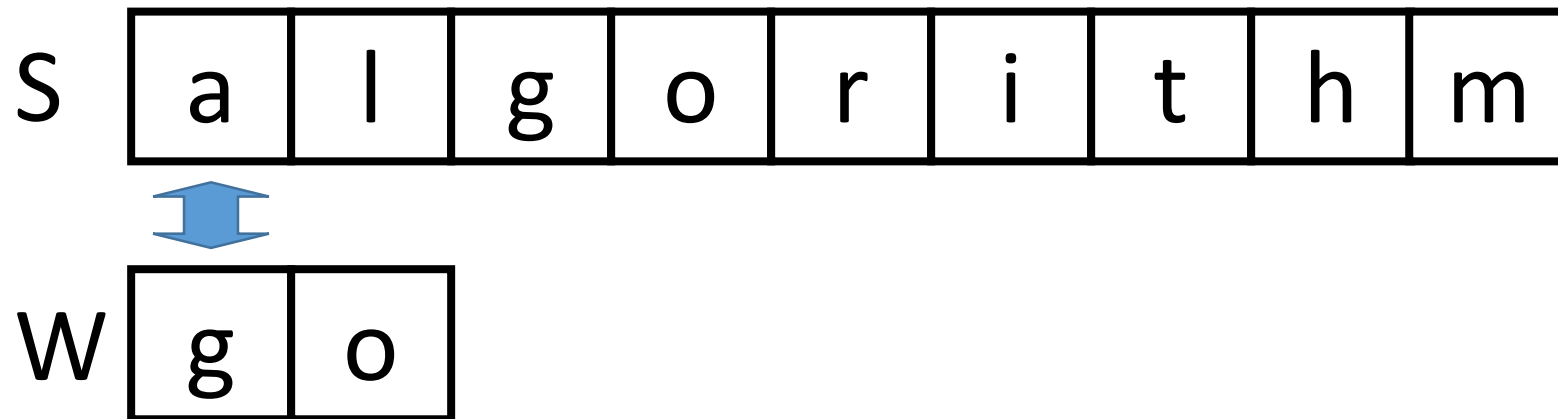
素朴な文字列照合アルゴリズム

Input: テキスト S , 単語 W

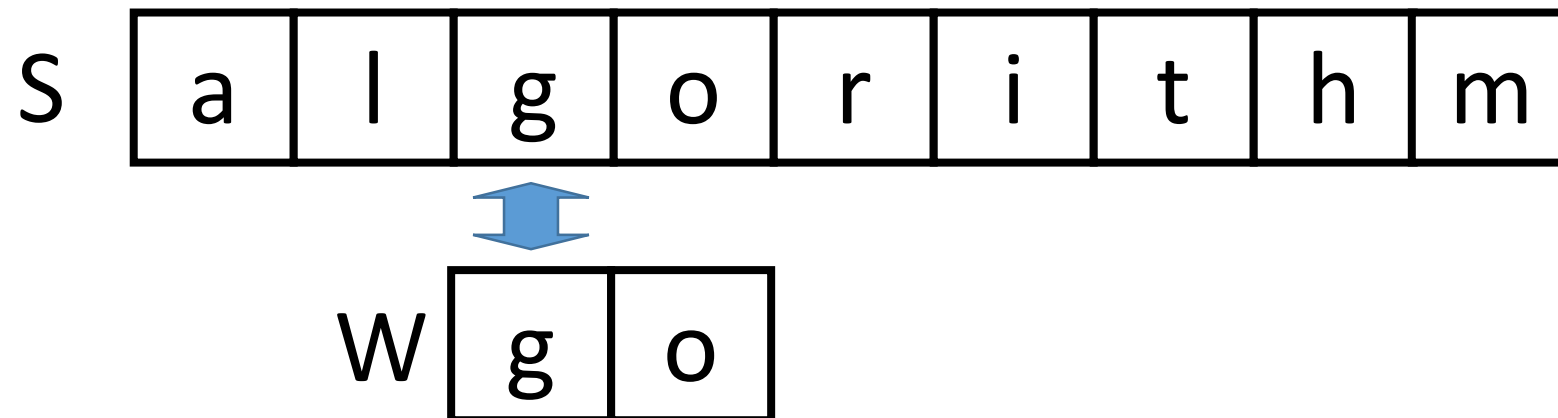
Output: S における W の最初の出現位置

1. $i \leftarrow 0; j \leftarrow 0$
2. while $i + j < |S|$ do
3. if $W[j] = S[i + j]$ then
4. $j \leftarrow j + 1$
5. もし $j = |W|$ なら i を出力して終了する
6. else
7. $i \leftarrow i + 1$
8. $j \leftarrow 0$
9. /* end of while */
10. $|S|$ を出力して終了する

$i = 0, j = 0$



$i = 2, j = 1$



都合の悪い例

S

a	a	a	b	a	a	a	b	a
---	---	---	---	---	---	---	---	---

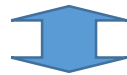


W

a	a	a	a
---	---	---	---

S

a	a	a	b	a	a	a	b	a
---	---	---	---	---	---	---	---	---



W

a	a	a	a
---	---	---	---

2. 3 KMP法

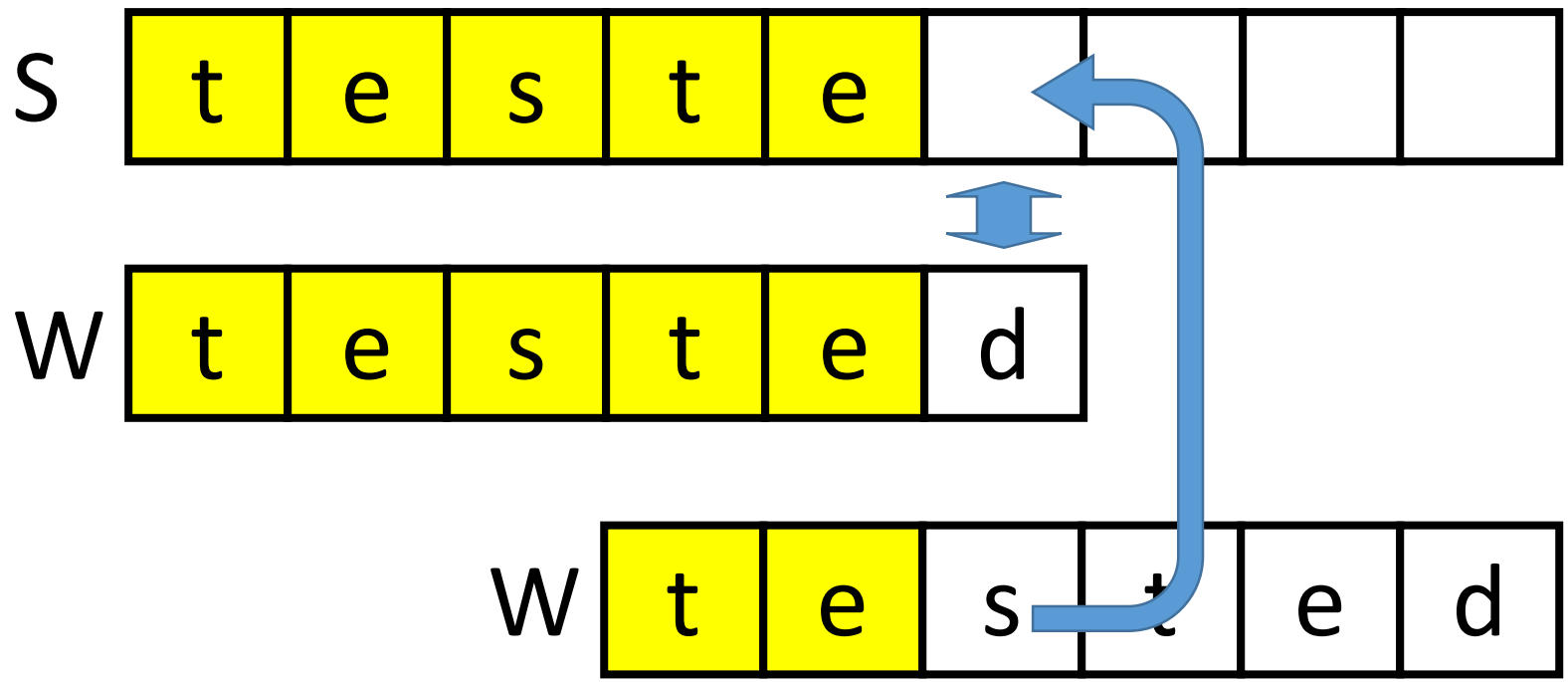
- クヌース-モリス-プラット法(Knuth-Morris-Pratt algorithm: KMP法)
- 照合に失敗したときの動作を改良した.
- W に依存したパターン照合テーブル(後述)を用いる.

KMP法

Input: テキストS, 単語W

Output: Wの出現位置

1. $i \leftarrow 0; j \leftarrow 0$
2. while $i + j < |S|$ do
3. if $W[j] = S[i + j]$ then
4. $j \leftarrow j + 1$
5. もし $j = |W|$ なら i を出力して終了
6. else
7. $i \leftarrow i + j - T[j]$
8. もし $j > 0$ なら $j \leftarrow T[j]$
9. /* end of while */



パターン照合テーブル

j	0	1	2	3	4	5
W	t	e	s	t	e	d
T	-1	0	0	0	2	0
$j - T[j]$	1	1	2	3	2	5

2. 4 パタン照合テーブルの構成法

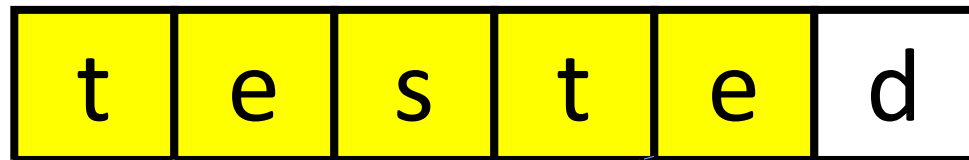
Input: 単語 W

Output: パタン照合テーブル T

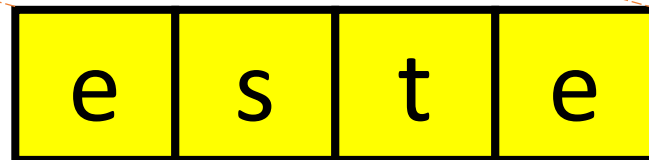
1. $T[0] = -1$
2. foreach $1 \leq j \leq |W| - 1$ do
3. $k \leftarrow j - 1$
4. while $W[0 .. k - 1] \neq W[j - k .. j - 1]$ and $k > 0$ do
5. $k \leftarrow k - 1$
6. /* end of while */
7. $T[j] \leftarrow k$
8. /* end of foreach */

$j = 4$

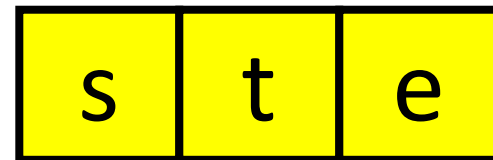
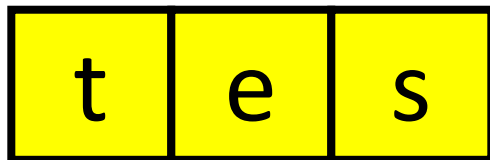
W



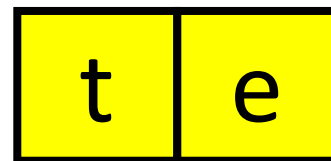
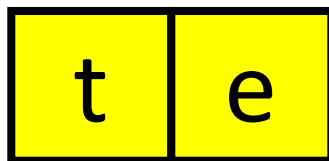
$k = 4$



$k = 3$



$k = 2$



評価

- パタン照合テーブルの作成には, $O(|W|^3)$ かかる.
- 改良により $O(|W|)$ で作成可能であることが知られている.