

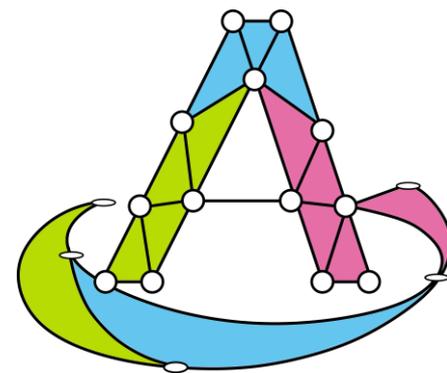
電子情報通信学会九州支部 2024年度（第32回）学生会講演会

変更制約付き最長共通部分列問題に 対する多項式時間アルゴリズム

◎ 高雄 奏摩* 新竹 優駿* 江藤 宏*
宮野 英次* 斎藤 寿樹* 塩田 拓海* **

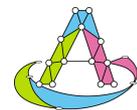
* 九州工業大学 ** JSPS 特別研究員

2024年 9月 25日 (水)



最長共通部分列

列 X の長さを $|X|$ と表記する



最長共通部分列 (LONGEST-COMMON SUBSEQUENCE, LCS)

入力 : 2つの文字列 $S = (s_0, s_1, \dots, s_{|S|-1})$, $T = (t_0, t_1, \dots, t_{|T|-1})$

目的 : S の部分列かつ T の部分列である共通部分列のうち
最長のものを求める

最長共通部分列問題の例

$S = a b c d e f$

$T = f c a e d e b f$

最長共通部分列

列 X の長さを $|X|$ と表記する



最長共通部分列 (LONGEST-COMMON SUBSEQUENCE, LCS)

入力 : 2つの文字列 $S = (s_0, s_1, \dots, s_{|S|-1})$, $T = (t_0, t_1, \dots, t_{|T|-1})$

目的 : S の部分列かつ T の部分列である共通部分列のうち
最長のものを求める

最長共通部分列問題の例

$S = a b c d e f$
 $T = f c a e d e b f$

✓ LCS : a d e f (LCS の長さ : 4)

最長共通部分列

列 X の長さを $|X|$ と表記する



最長共通部分列 (LONGEST-COMMON SUBSEQUENCE, LCS)

入力 : 2つの文字列 $S = (s_0, s_1, \dots, s_{|S|-1})$, $T = (t_0, t_1, \dots, t_{|T|-1})$

目的 : S の部分列かつ T の部分列である共通部分列のうち
最長のものを求める

最長共通部分列問題の例

$S = a b c d e f$
 $T = f c a e d e b f$

✓ LCS : a d e f (LCS の長さ : 4)

LCS の長さを求める最適化問題を **LCS 問題** という

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

BD-LCS問題の例

$S = a b c d e e a b c d$

$T = e a b c d a c d b e$

$Z = e e \quad A = (4, 5)$

$B = (0, 9) \quad k = 1$

変更制約付き最長共通部分列問題



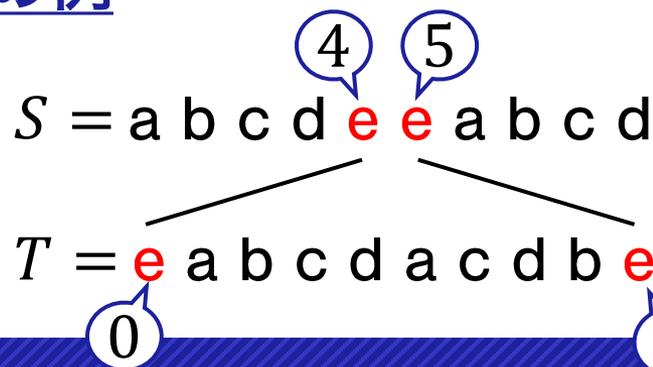
変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する
添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の
長さを求める

BD-LCS問題の例



$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

BD-LCS問題の例

$S = a b c d e e a b c d$
 $T = e a b c d a c d b e$

Diagram showing the alignment of the strings S and T. The character 'e' at index 4 of S is aligned with the character 'e' at index 0 of T. A line connects the 'e' at index 4 of S to the 'e' at index 0 of T. A speech bubble with the number '4' is above the 'e' in S, and a speech bubble with the number '0' is below the 'e' in T.

$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

BD-LCS問題の例

$S = a b c d e e a b c d$
 $T = e a b c d a c d b e$

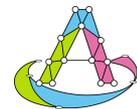
④
①

$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

共通部分列の長さ = 4

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

BD-LCS問題の例

$S = a b c d e e a b c d$

$T = e a b c d a c d b e$

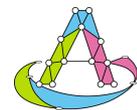
5

9

$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

BD-LCS問題の例

$S = a b c d e e a b c d$
 $T = e a b c d a c d b e$

$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

共通部分列の長さ = 5

9

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

BD-LCS問題の例

解 = 5

$S = a b c d e e a b c d$
 $T = e a b c d a c d b e$

$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

共通部分列の長さ = 5

9

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

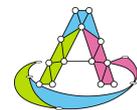
条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

本研究の成果

BD-LCS 問題に対し $O(|S||T||Z|)$ の計算時間量で, 解を求める多項式時間アルゴリズムを示した.

LCS 問題に対する既存アルゴリズム



[R. A. Wanger et al., 1974]

2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$

		e	a	b	c	d
	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)
a	(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
c	(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
d	(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
b	(4,0)	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
e	(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

LCS 問題に対する既存アルゴリズム



[R. A. Wanger et al., 1974]

2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$

	e	a	b	c	d	
	(0,0)	(0,1)	(0,2)	(0,3)	(0,4)	(0,5)
a	(1,0)					
c	(2,0)	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
d	(3,0)	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
b	(4,0)	(4,1)	(4,2)			
e	(5,0)	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

S の 0 文字目までと
T の 1 文字目までのLCS

S の 2 文字目までと
T の 4 文字目までのLCS

LCS 問題に対する既存アルゴリズム



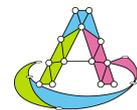
[R. A. Wanger et al., 1974]

2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$

	e	a	b	c	d
	(0,0) → (0,1) → (0,2) → (0,3) → (0,4) → (0,5)				
a	(1,0) → (1,1) → (1,2) → (1,3) → (1,4) → (1,5)				
c	(2,0) → (2,1) → (2,2) → (2,3) → (2,4) → (2,5)				
d	(3,0) → (3,1) → (3,2) → (3,3) → (3,4) → (3,5)				
b	(4,0) → (4,1) → (4,2) → (4,3) → (4,4) → (4,5)				
e	(5,0) → (5,1) → (5,2) → (5,3) → (5,4) → (5,5)				

LCS 問題に対する既存アルゴリズム



[R. A. Wanger et al., 1974]

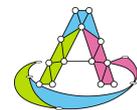
2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$

	e	a	b	c	d
	(0,0) → (0,1) → (0,2) → (0,3) → (0,4) → (0,5)				
a	(1,0) → (1,1) → (1,2) → (1,3) → (1,4) → (1,5)				
c	(2,0) → (2,1) → (2,2) → (2,3) → (2,4) → (2,5)				
d	(3,0) → (3,1) → (3,2) → (3,3) → (3,4) → (3,5)				
b	(4,0) → (4,1) → (4,2) → (4,3) → (4,4) → (4,5)				
e	(5,0) → (5,1) → (5,2) → (5,3) → (5,4) → (5,5)				

対応する
文字が同じ

LCS 問題に対する既存アルゴリズム



[R. A. Wanger et al., 1974]

2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$

	e	a	b	c	d
	(0,0) → (0,1) → (0,2) → (0,3) → (0,4) → (0,5)				
a	(1,0) → (1,1) → (1,2) → (1,3) → (1,4) → (1,5)				
c	(2,0) → (2,1) → (2,2) → (2,3) → (2,4) → (2,5)				
d	(3,0) → (3,1) → (3,2) → (3,3) → (3,4) → (3,5)				
b	(4,0) → (4,1) → (4,2) → (4,3) → (4,4) → (4,5)				
e	(5,0) → (5,1) → (5,2) → (5,3) → (5,4) → (5,5)				

共通部分列
ab に対応

LCS 問題に対する既存アルゴリズム



[R. A. Wanger et al., 1974]

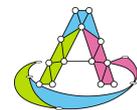
2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$

	e	a	b	c	d
	(0,0) → (0,1)	(0,1) → (0,2)	(0,2) → (0,3)	(0,3) → (0,4)	(0,4) → (0,5)
a	(1,0) → (1,1)	(1,1) → (1,2)	(1,2) → (1,3)	(1,3) → (1,4)	(1,4) → (1,5)
c	(2,0) → (2,1)	(2,1) → (2,2)	(2,2) → (2,3)	(2,3) → (2,4)	(2,4) → (2,5)
d	(3,0) → (3,1)	(3,1) → (3,2)	(3,2) → (3,3)	(3,3) → (3,4)	(3,4) → (3,5)
b	(4,0) → (4,1)	(4,1) → (4,2)	(4,2) → (4,3)	(4,3) → (4,4)	(4,4) → (4,5)
e	(5,0) → (5,1)	(5,1) → (5,2)	(5,2) → (5,3)	(5,3) → (5,4)	(5,4) → (5,5)

共通部分列
acd に対応

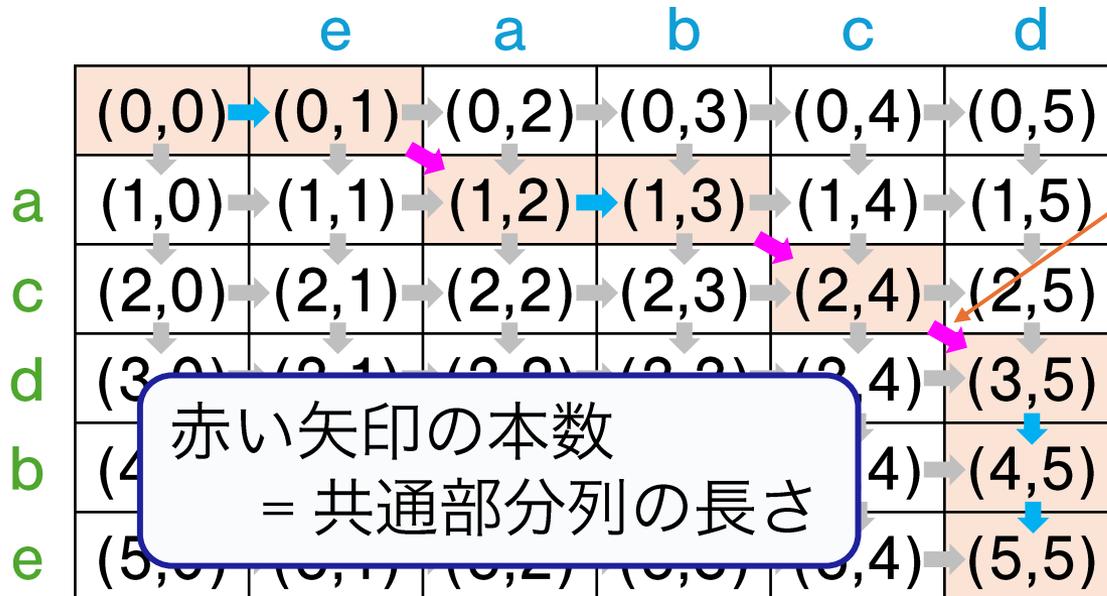
LCS 問題に対する既存アルゴリズム



[R. A. Wanger et al., 1974]

2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$



共通部分列
acd に対応

LCS 問題に対する既存アルゴリズム



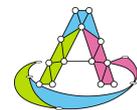
[R. A. Wanger et al., 1974]

2つの文字列 S, T に対して, $O(|S||T|)$ の計算時間量で LCS を求めることができる.

例 $S = e a b c d$, $T = a c d b e$



LCS 問題に対する既存アルゴリズム

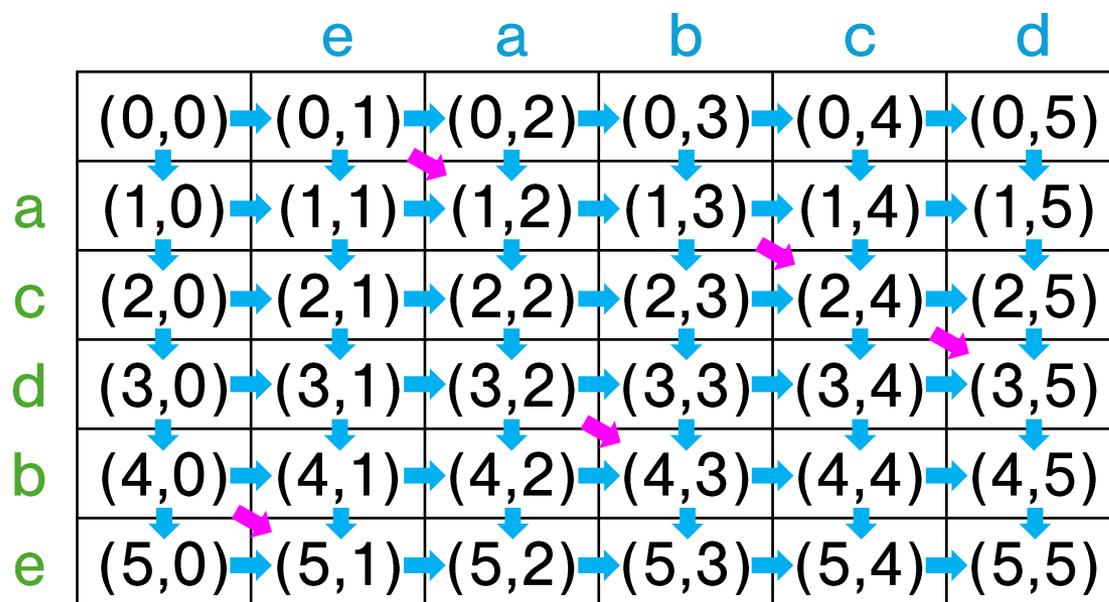


赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム

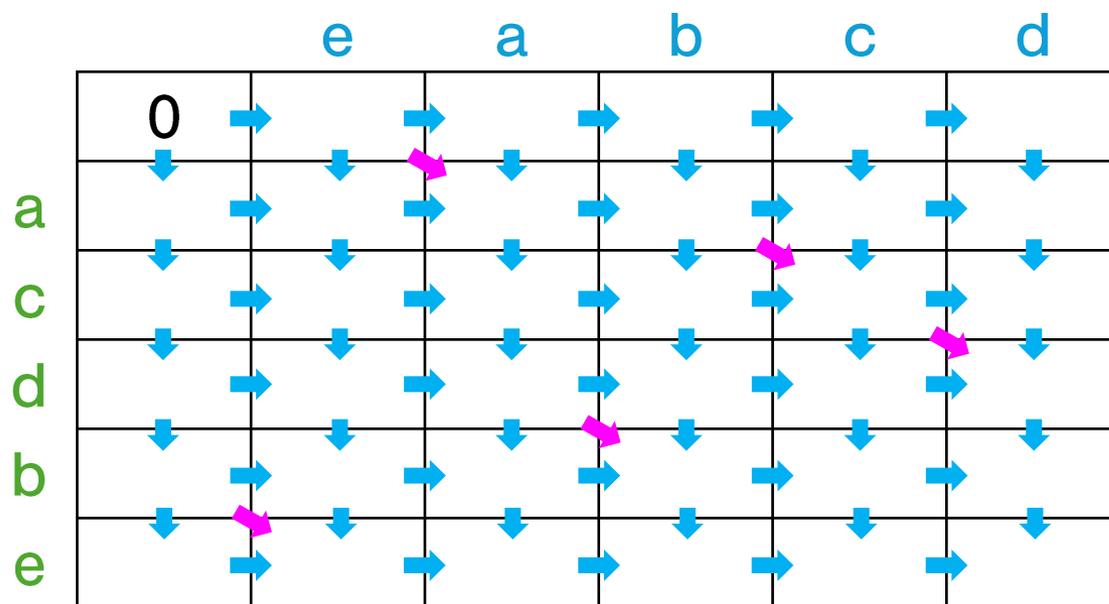


赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



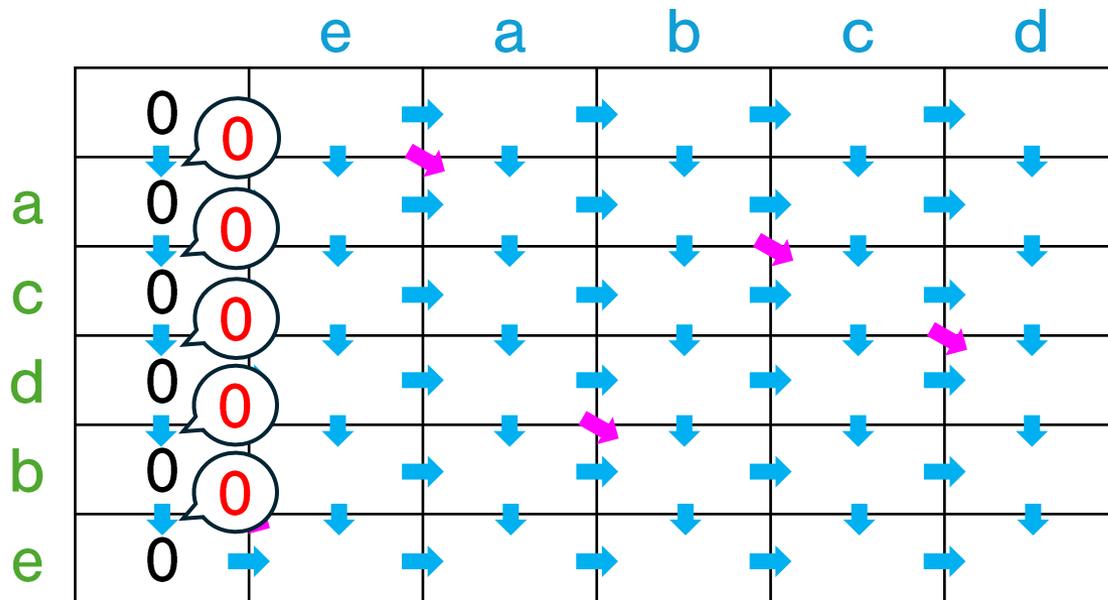
LCS 問題に対する既存アルゴリズム



赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

- 移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印
③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム

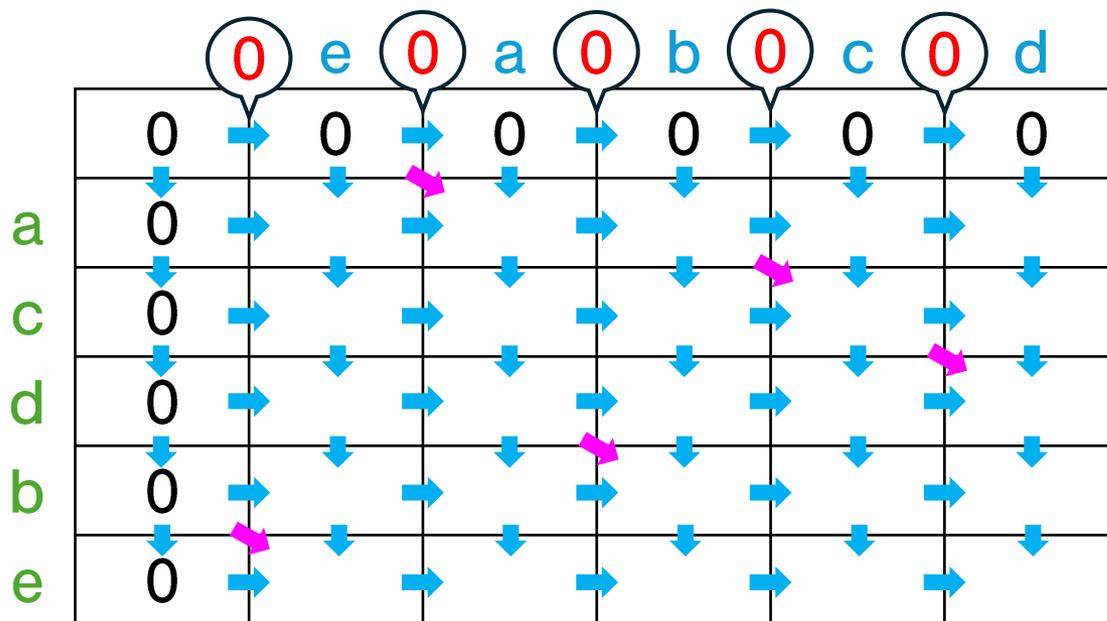


赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム

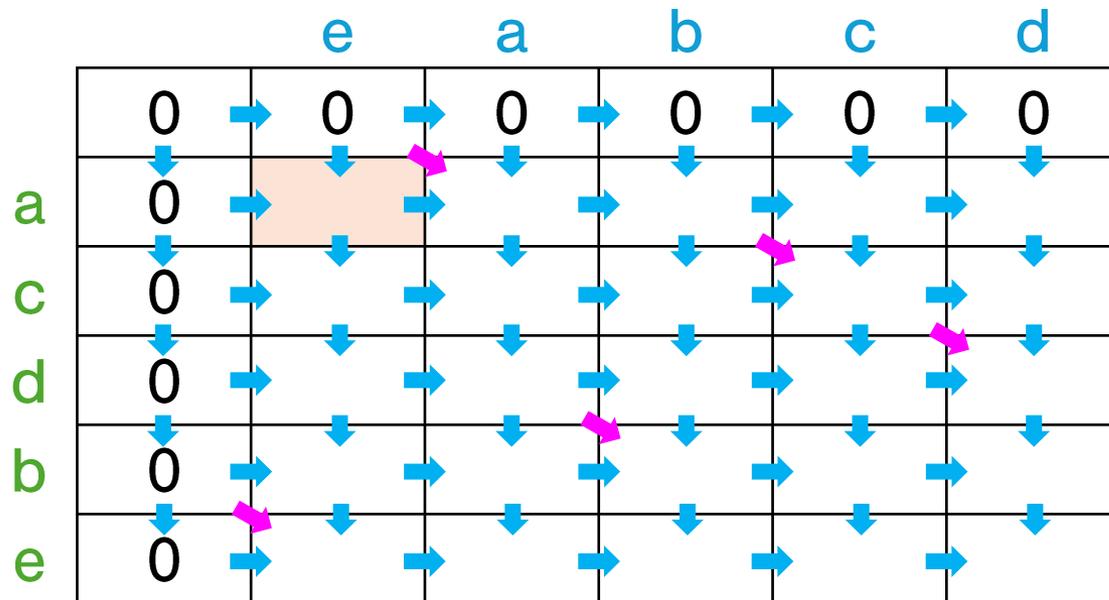


赤い矢印の最大本数を求めるための動的計画法を考える

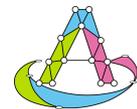
$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム

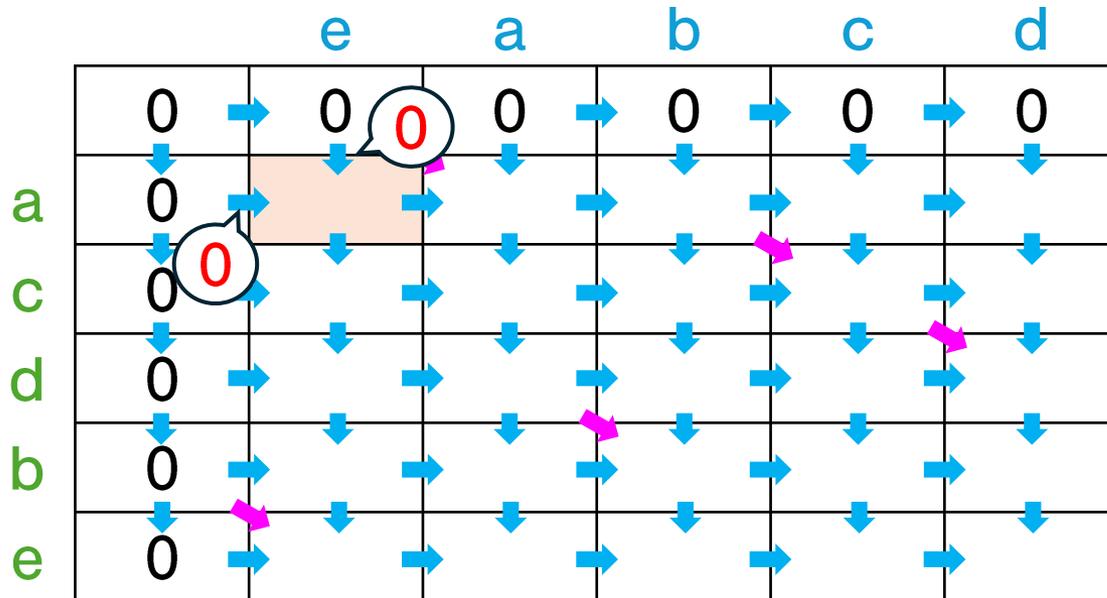


赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム

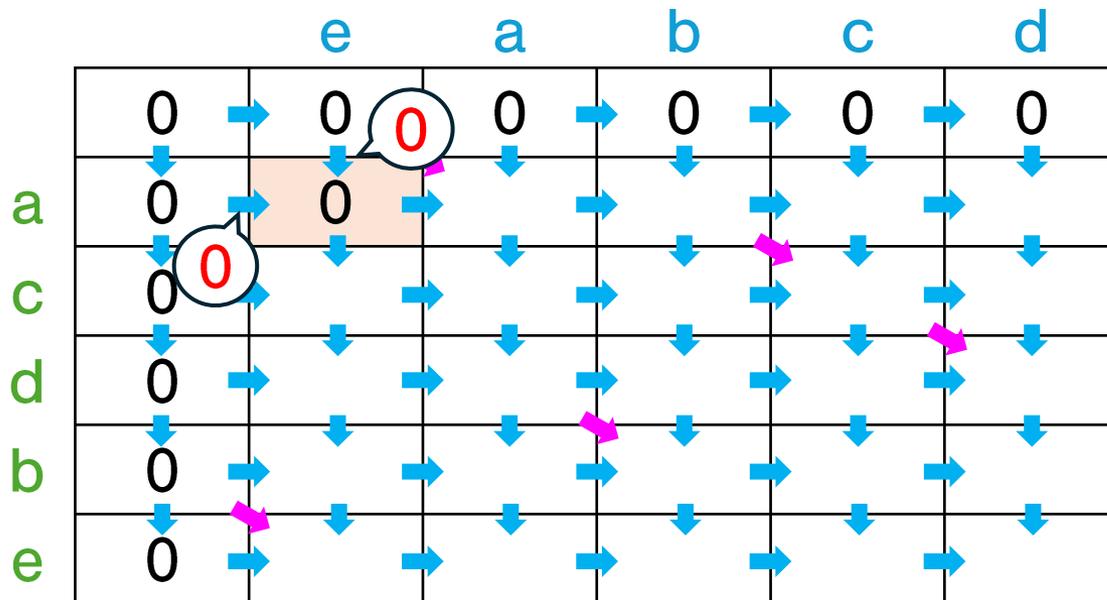


赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム



赤い矢印の最大本数を求めるための動的計画法を考える

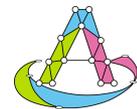
$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

- 移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印
③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	0	0	0	0
c	0	0	0	0	0	0
d	0	0	0	0	0	0
b	0	0	0	0	0	0
e	0	0	0	0	0	0

The diagram shows a 6x6 grid of DP values. The top row and left column are labeled with '0'. The columns are labeled 'e', 'a', 'b', 'c', 'd' and the rows are labeled 'a', 'c', 'd', 'b', 'e'. Blue arrows indicate transitions from (i-1, j) to (i, j) (down) and from (i, j-1) to (i, j) (right). Pink arrows indicate transitions from (i-1, j-1) to (i, j) (diagonal), which are the red arrows mentioned in the text. A light orange shaded area covers the cells (a, a), (a, b), (c, a), and (c, b).

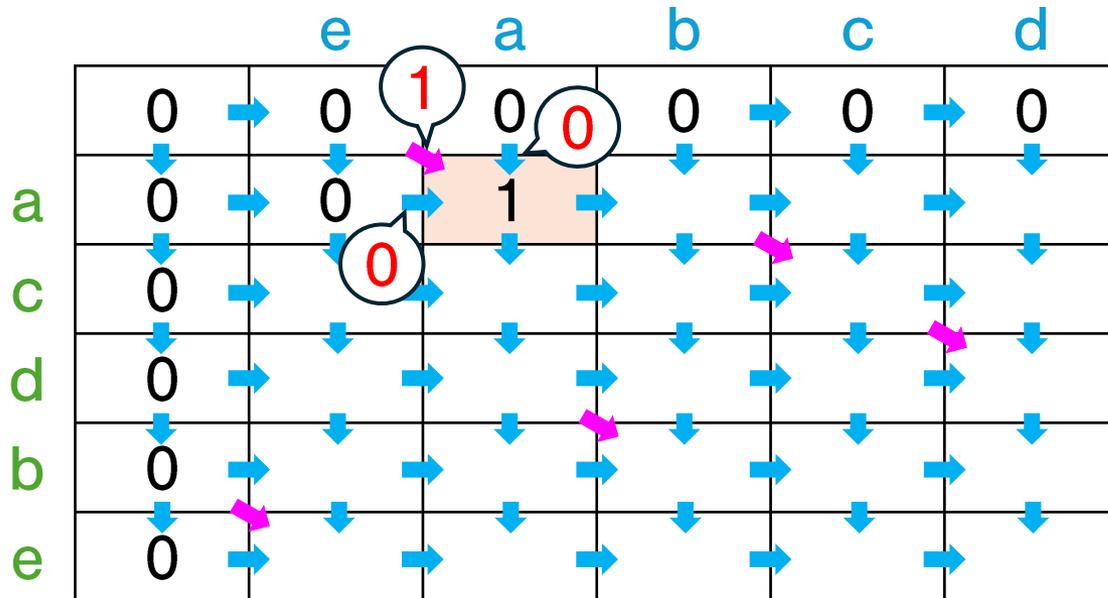
LCS 問題に対する既存アルゴリズム



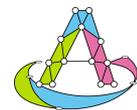
赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

- 移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印
③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)



LCS 問題に対する既存アルゴリズム



赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

- 移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印
③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	1	1	1	1
c	0	0	1	1	2	2
d	0	0	1	1	2	3
b	0	0	1	2	2	3
e	0	1	1	2	2	3

LCS 問題に対する既存アルゴリズム



赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

- 移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印
③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	1	1	1	1
c	0	0	1	1	2	2
d	0	0	1	1	2	3
b	0	0	1	2	2	3

テーブルを埋めるのに $(|S| + 1)(|T| + 1)$ ステップが必要

LCS 問題に対する既存アルゴリズム



赤い矢印の最大本数を求めるための動的計画法を考える

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値

- 移動方法: ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印
③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	1	1	1	1
c	0	0	1	1	2	2
d	0	0	1	1	2	3
b	0	0	1	2	2	3

$O(|S||T|)$

テーブルを埋めるのに $(|S| + 1)(|T| + 1)$ ステップが必要

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

削除する組みの組合せを 2^k 通り考えないといけない?

BD-LCS問題の例

解 = 5

$S = a b c d e e a b c d$
 $T = e a b c d a c d b e$

Diagram showing alignment of S and T with lines connecting 'e' at index 4 of S to 'e' at index 9 of T, and 'e' at index 5 of S to 'e' at index 0 of T. A circled '5' is above the first 'e' in S, and a circled '9' is below the 'e' at index 9 in T.

$Z = e e$ $A = (4, 5)$

$B = (0, 9)$ $k = 1$

共通部分列の長さ = 5

9

変更制約付き最長共通部分列問題



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する
添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: ~~k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の
長さを求める~~



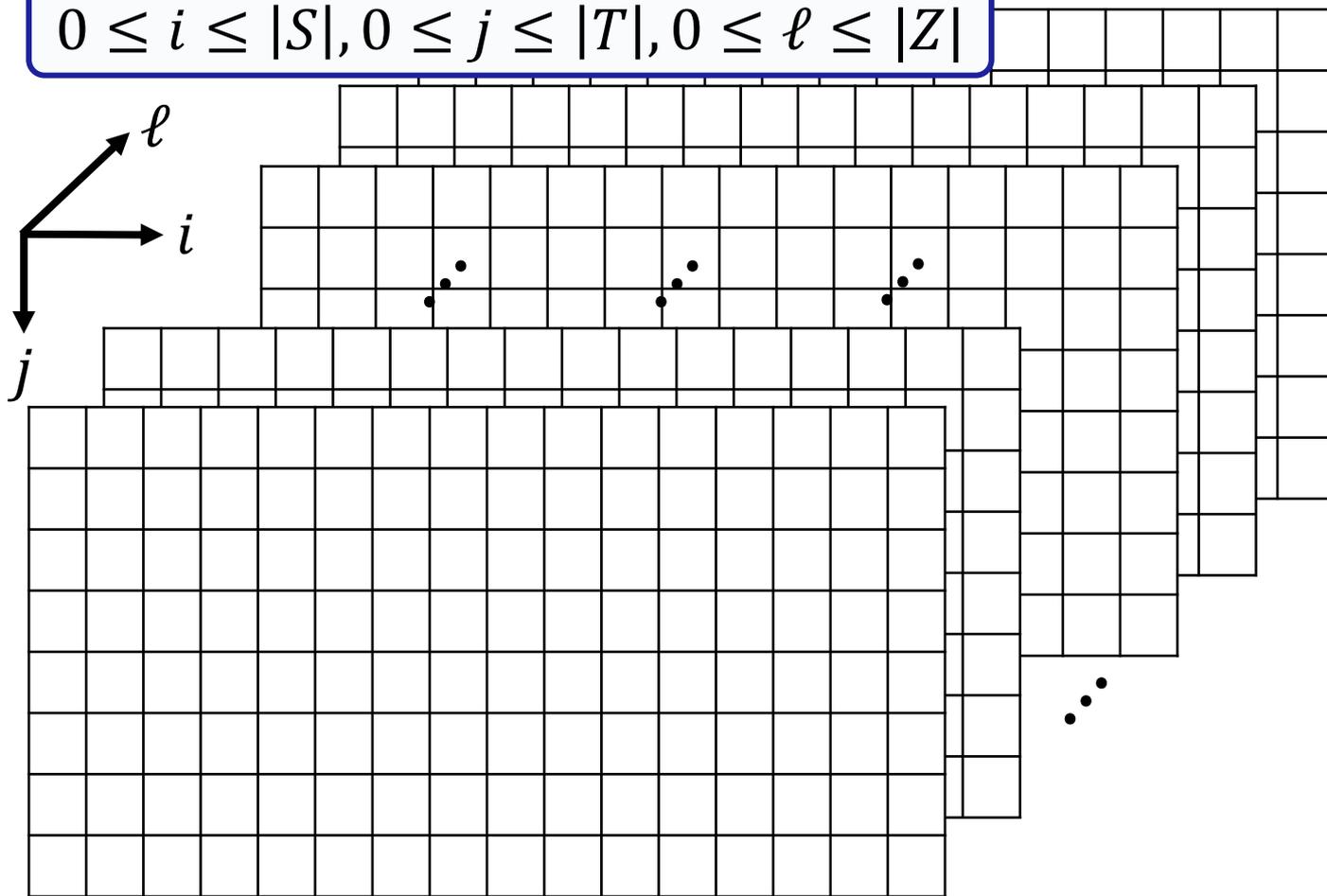
$(|Z| - k)$ 組み以上の $(S[a_i], T[b_i])$ からなる, LCS の
長さを求める

BD-LCS 問題に対するアルゴリズム

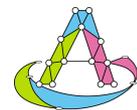


$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$$

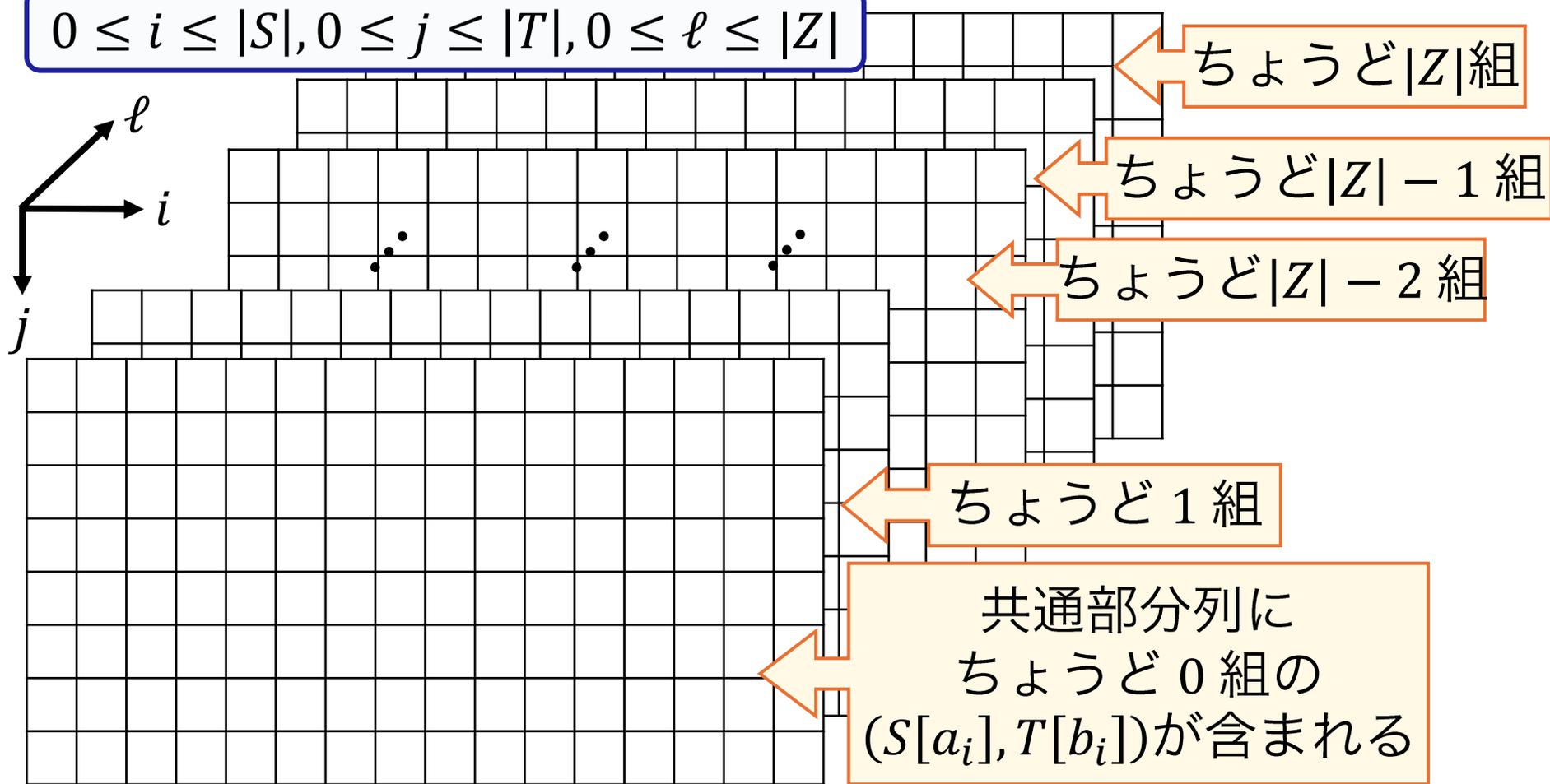


BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$$

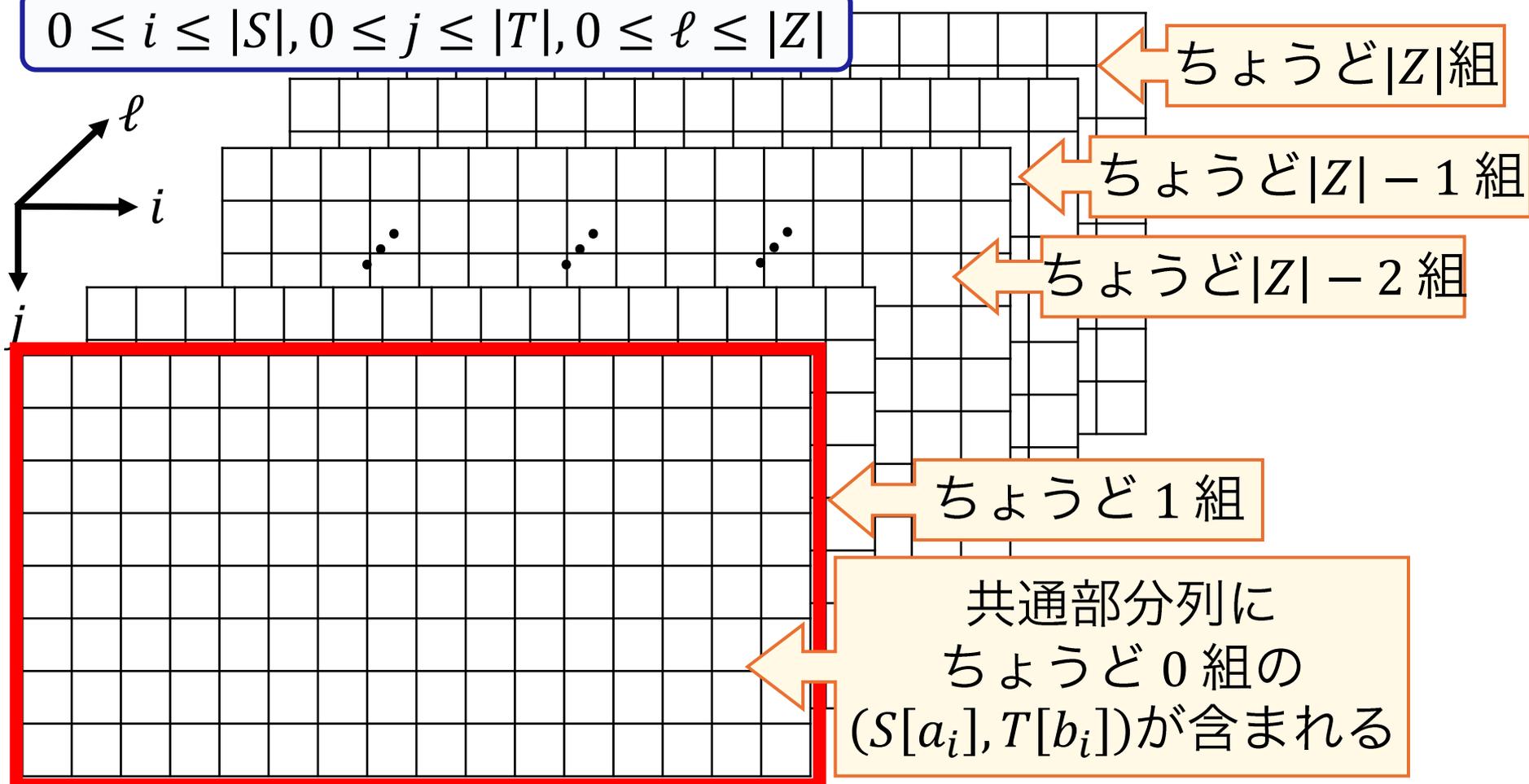


BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$$



BD-LCS 問題に対するアルゴリズム



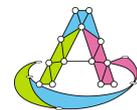
例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

条件 共通部分列にちょうど 0 組の $(S[a_i], T[b_i])$ が含まれる

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値
($dp[i][j][\ell]$ の ℓ の値は 0 であるため省略)

	e	a	b	c	d
	(0,0) → (0,1) → (0,2) → (0,3) → (0,4) → (0,5)				
a	(1,0) → (1,1) → (1,2) → (1,3) → (1,4) → (1,5)				
c	(2,0) → (2,1) → (2,2) → (2,3) → (2,4) → (2,5)				
d	(3,0) → (3,1) → (3,2) → (3,3) → (3,4) → (3,5)				
b	(4,0) → (4,1) → (4,2) → (4,3) → (4,4) → (4,5)				
e	(5,0) → (5,1) → (5,2) → (5,3) → (5,4) → (5,5)				

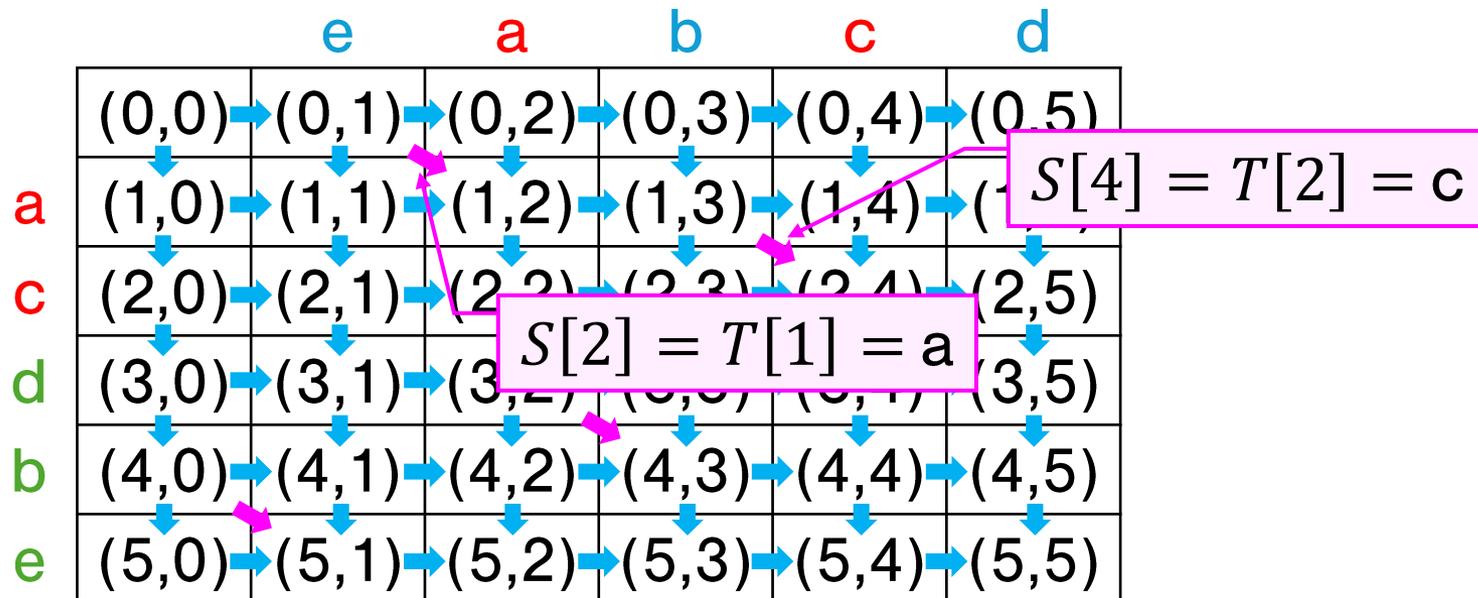
BD-LCS 問題に対するアルゴリズム



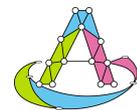
例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

条件 共通部分列にちょうど 0 組の $(S[a_i], T[b_i])$ が含まれる

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値
($dp[i][j][\ell]$ の ℓ の値は 0 であるため省略)



BD-LCS 問題に対するアルゴリズム



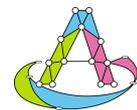
例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

条件 共通部分列にちょうど 0 組の $(S[a_i], T[b_i])$ が含まれる

$dp[i][j]$: マス (i, j) に到達するまでに通る赤い矢印の最大値
($dp[i][j][\ell]$ の ℓ の値は 0 であるため省略)

	e	a	b	c	d
	(0,0) → (0,1) → (0,2) → (0,3) → (0,4) → (0,5)				
a	(1,0) → (1,1) → (1,2) → (1,3) → (1,4) → (1,5)				
c	(2,0) → (2,1) → (2,2) → (2,3) → (2,4) → (2,5)				
d	(3,0) → (3,1) → (3,2) → (3,3) → (3,4) → (3,5)				
b	(4,0) → (4,1) → (4,2) → (4,3) → (4,4) → (4,5)				
e	(5,0) → (5,1) → (5,2) → (5,3) → (5,4) → (5,5)				

BD-LCS 問題に対するアルゴリズム



例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

条件 共通部分列にちょうど 0 組の $(S[a_i], T[b_i])$ が含まれる

移動方法 : ① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	0	0	0	0
c	0	0	0	0	0	0
d	0	0	0	0	0	1
b	0	0	0	1	1	1
e	0	1	1	1	1	1

BD-LCS 問題に対するアルゴリズム

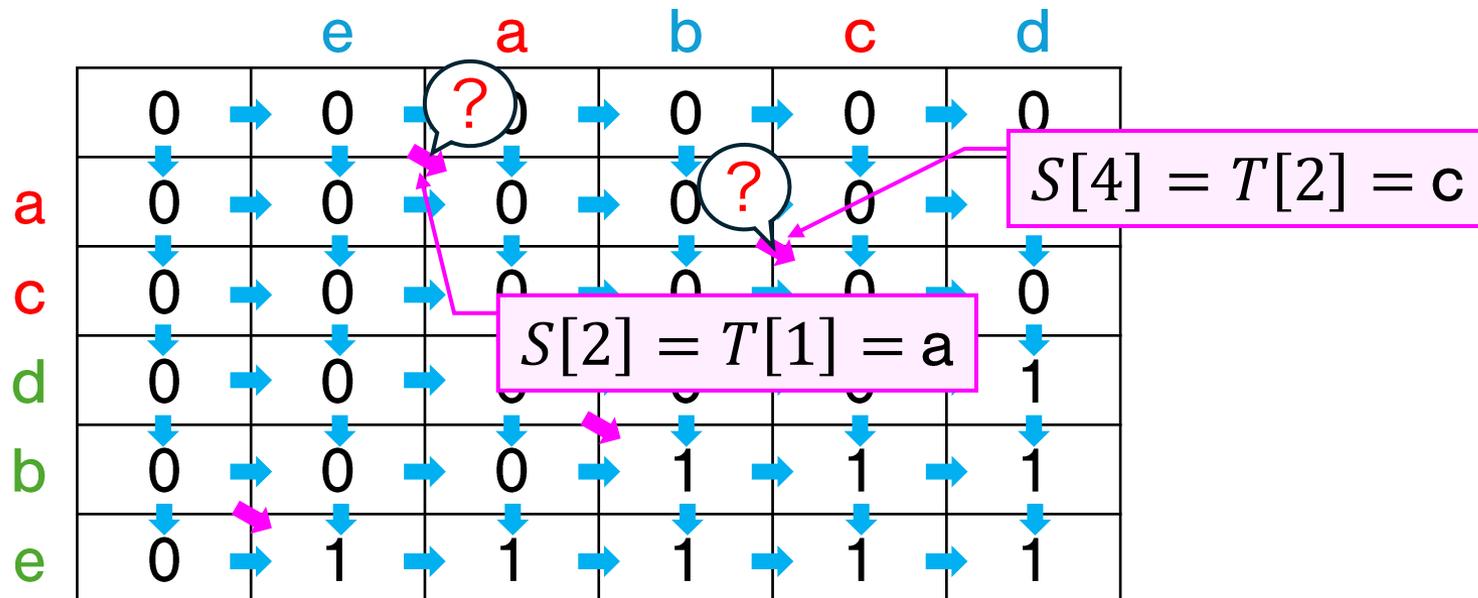


例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

条件 共通部分列にちょうど 0 組の $(S[a_i], T[b_i])$ が含まれる

移動方法：① $(i-1, j)$ から下矢印 ② $(i, j-1)$ から右矢印

③ $(i-1, j-1)$ から赤い矢印 (値を +1 する)

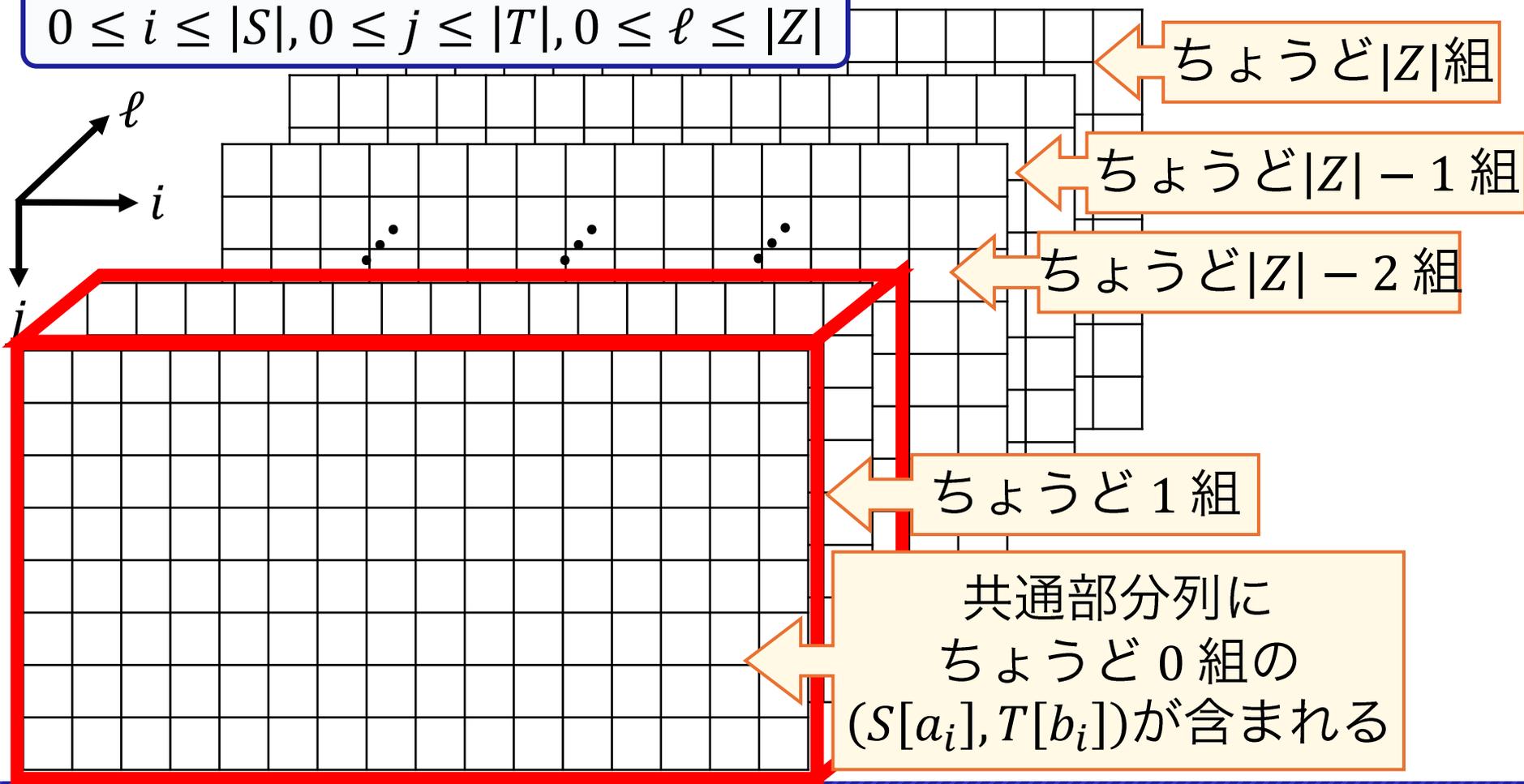


BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$$



BD-LCS 問題に対するアルゴリズム

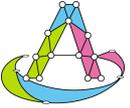


例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	0	0	0	0
c	0	0	0	0	0	0
d	0	0	0	0	0	1
b	0	0	0	1	1	1
e	0	1	1	1	1	1

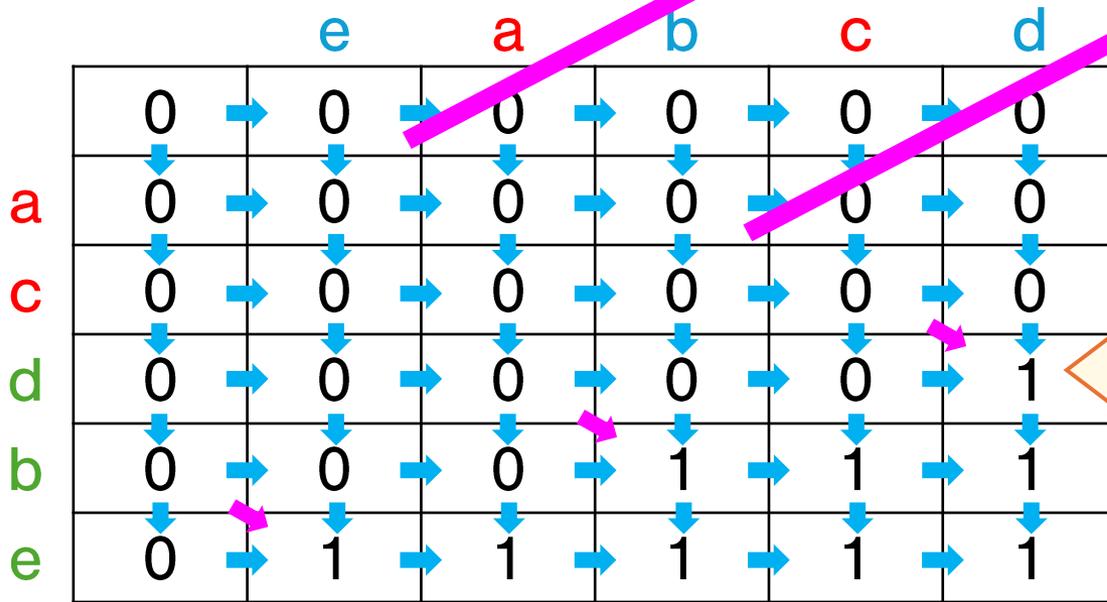
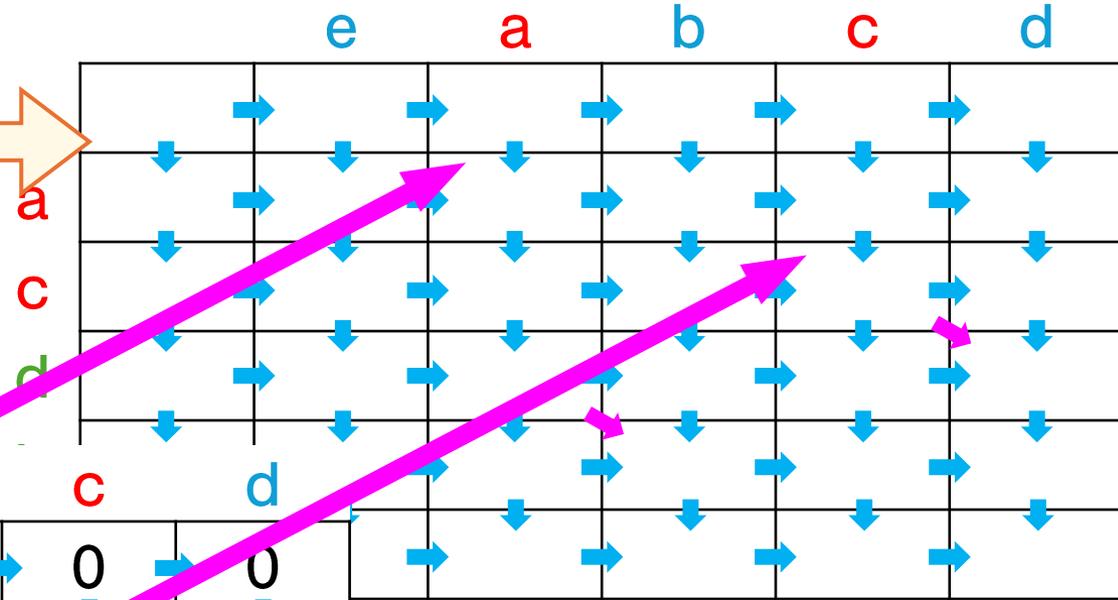
共通部分列に
ちょうど 0 組の
($S[a_i], T[b_i]$)が含まれる

BD-LCS 問題に対するアルゴリズム



例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
 $(S[a_i], T[b_i])$ が含まれる



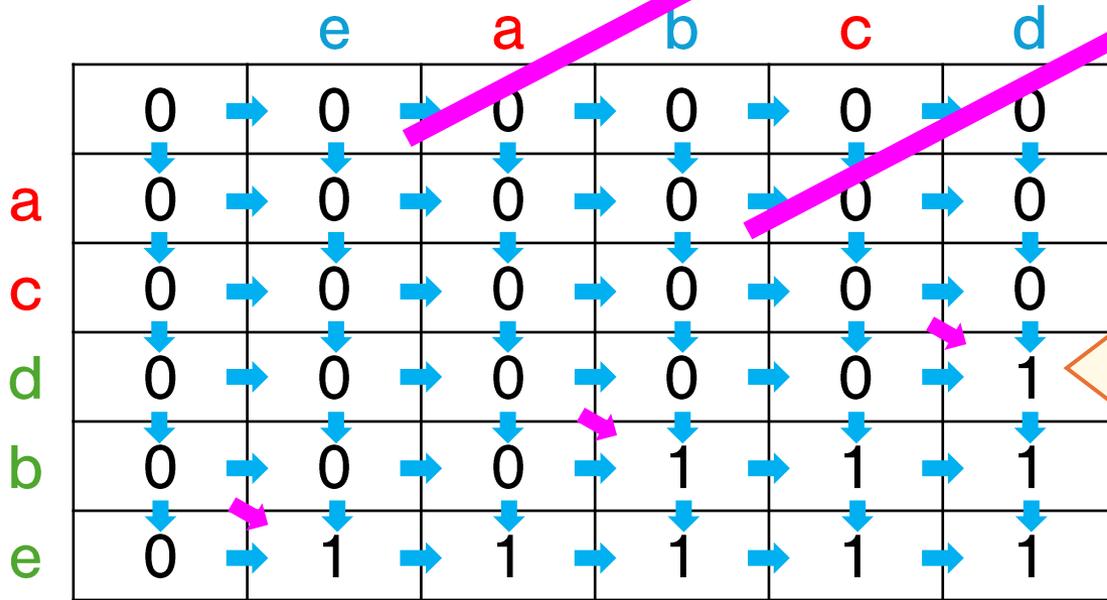
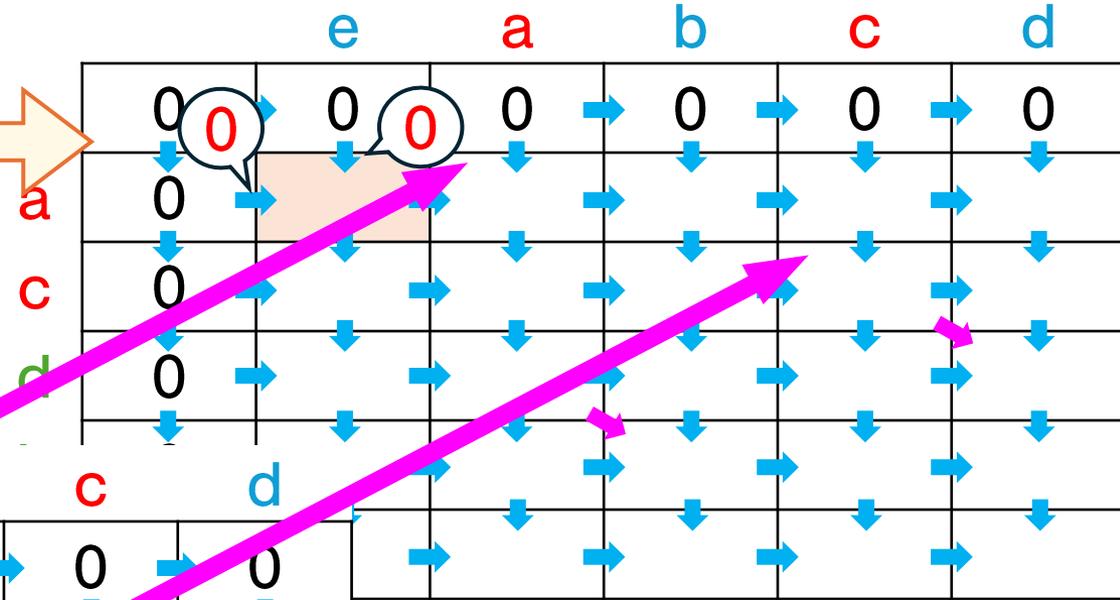
共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

BD-LCS 問題に対するアルゴリズム



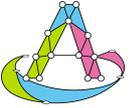
例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
 $(S[a_i], T[b_i])$ が含まれる



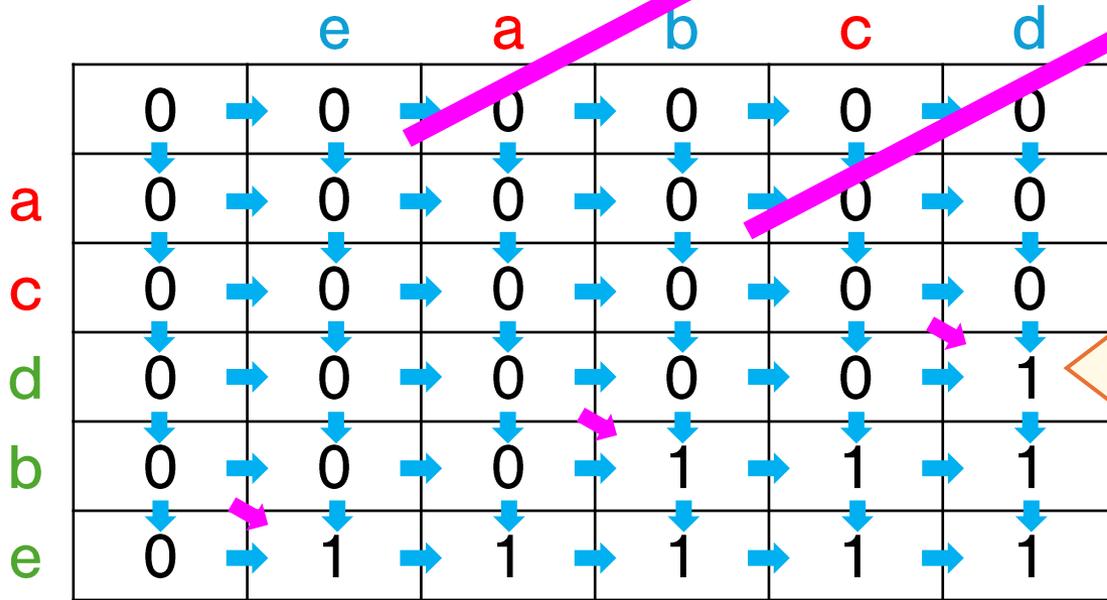
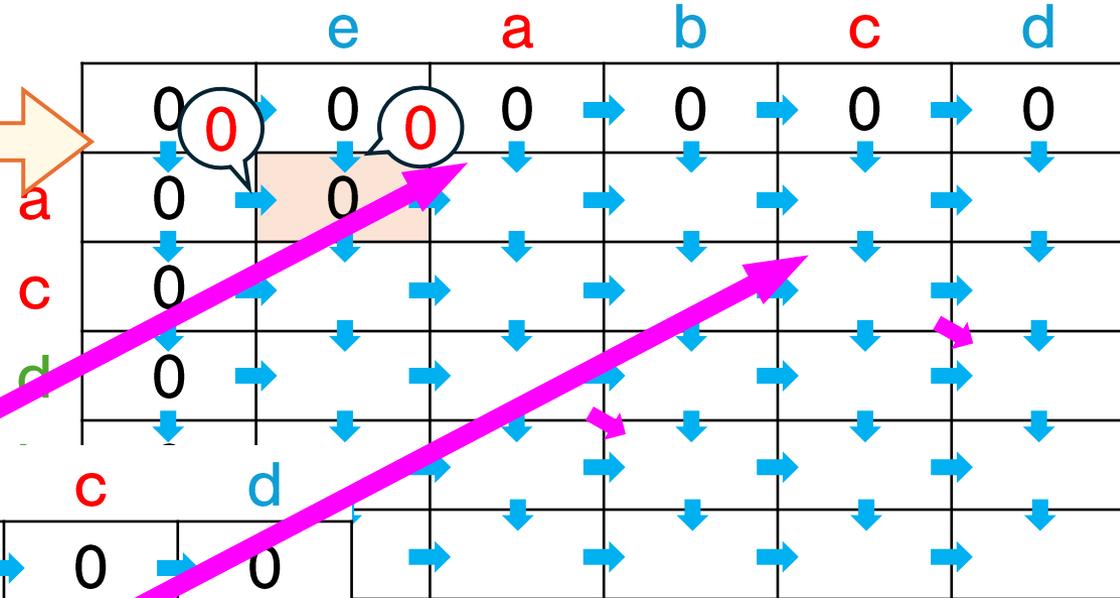
共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

BD-LCS 問題に対するアルゴリズム



例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
 $(S[a_i], T[b_i])$ が含まれる



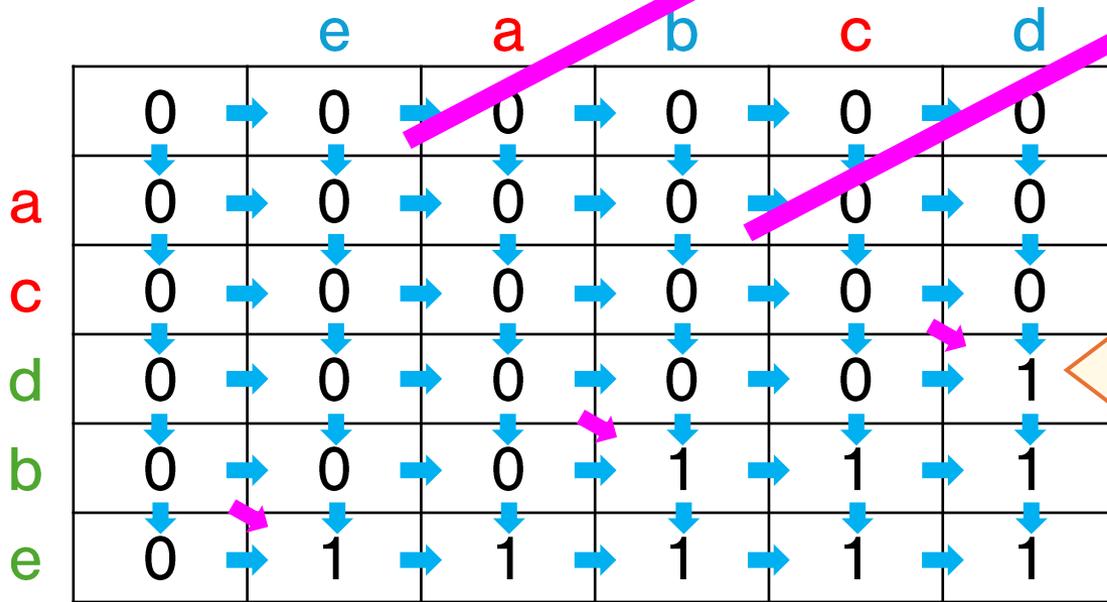
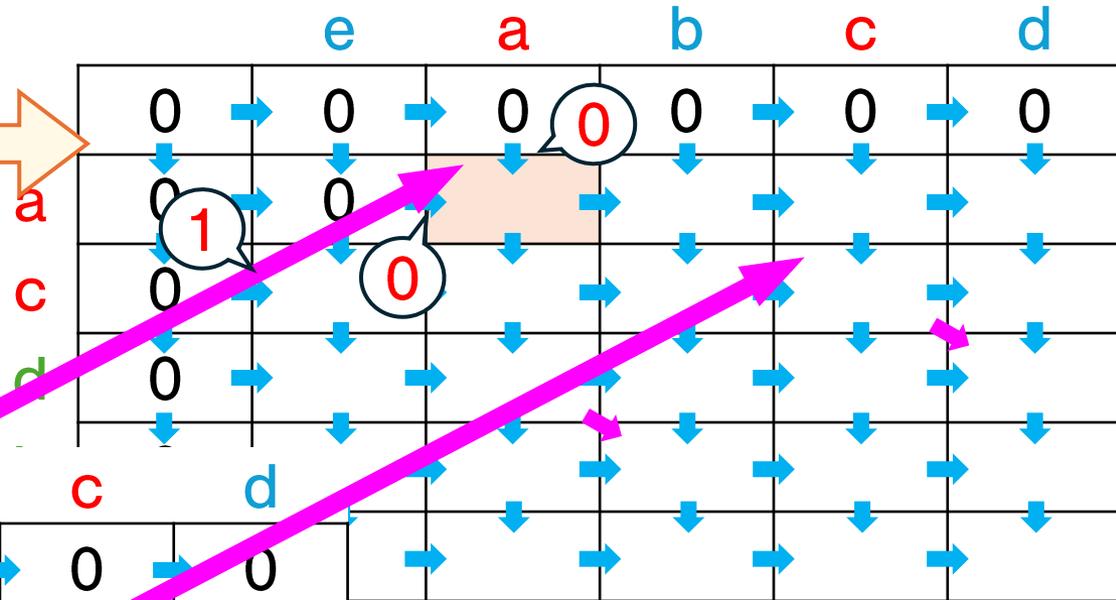
共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

BD-LCS 問題に対するアルゴリズム



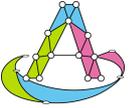
例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
($S[a_i], T[b_i]$)が含まれる



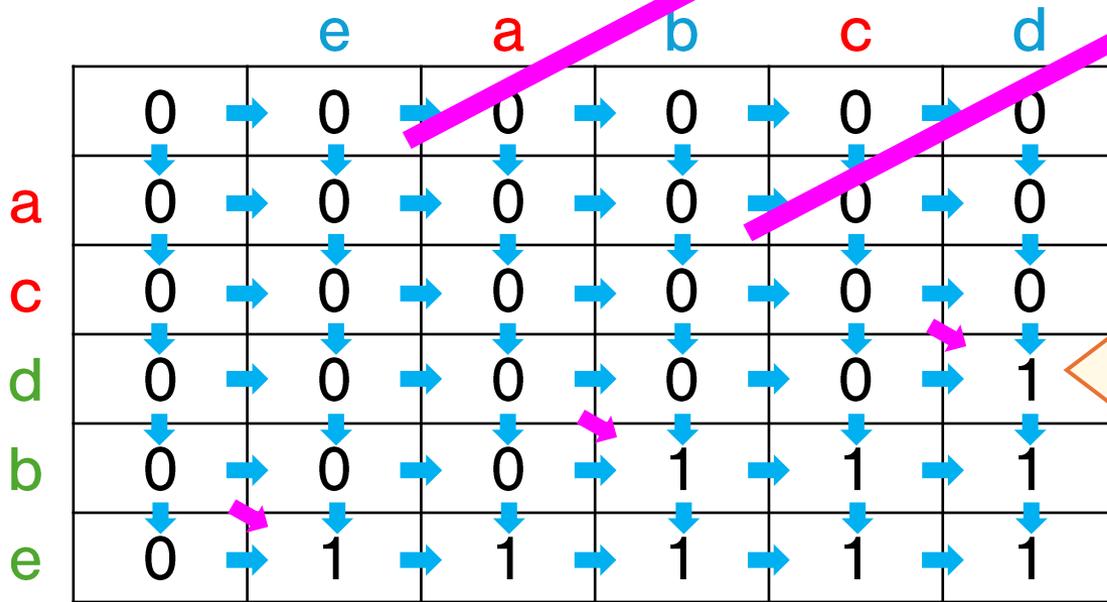
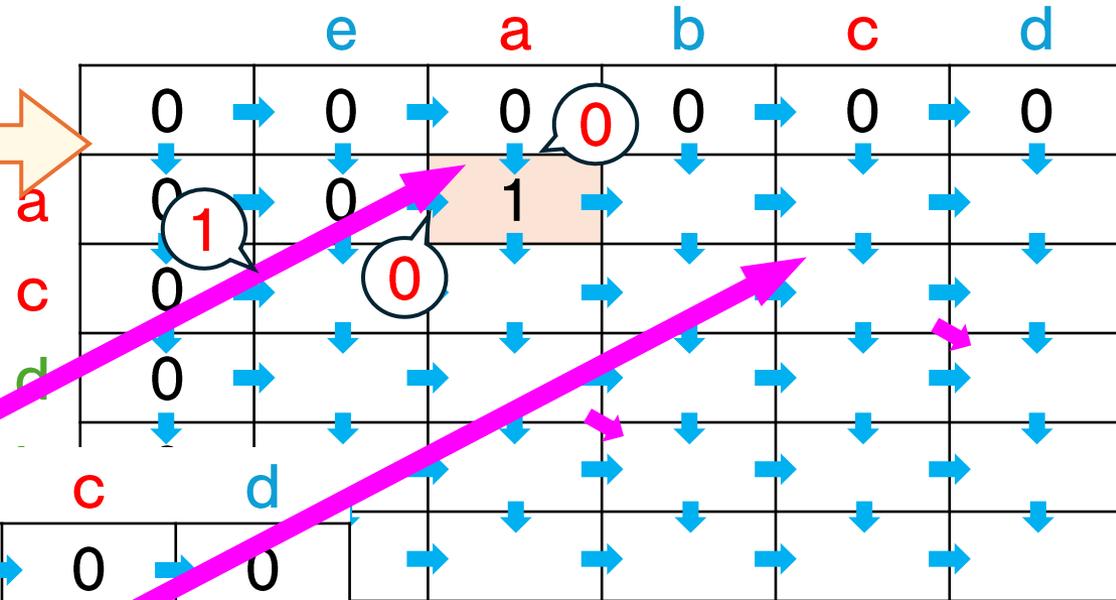
共通部分列に
ちょうど 0 組の
($S[a_i], T[b_i]$)が含まれる

BD-LCS 問題に対するアルゴリズム



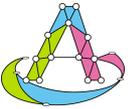
例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
 $(S[a_i], T[b_i])$ が含まれる



共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

BD-LCS 問題に対するアルゴリズム



例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
 $(S[a_i], T[b_i])$ が含まれる

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	1	1	1	1
c	0	0	1	1	1	1
d	0	0	1	1	1	1
e	0	0	1	2	2	2

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	0	0	0	0
c	0	0	0	0	0	0
d	0	0	0	0	0	1
b	0	0	0	1	1	1
e	0	1	1	1	1	1

共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

BD-LCS 問題に対するアルゴリズム



例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$

共通部分列に
ちょうど 1 組の
 $(S[a_i], T[b_i])$ が含まれる

		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	1	1	1	1
c	0	0	1	1	1	1
d	0	0	1	1	1	2
b	0	0	1	2	2	2
e	0	1	2	2	2	2

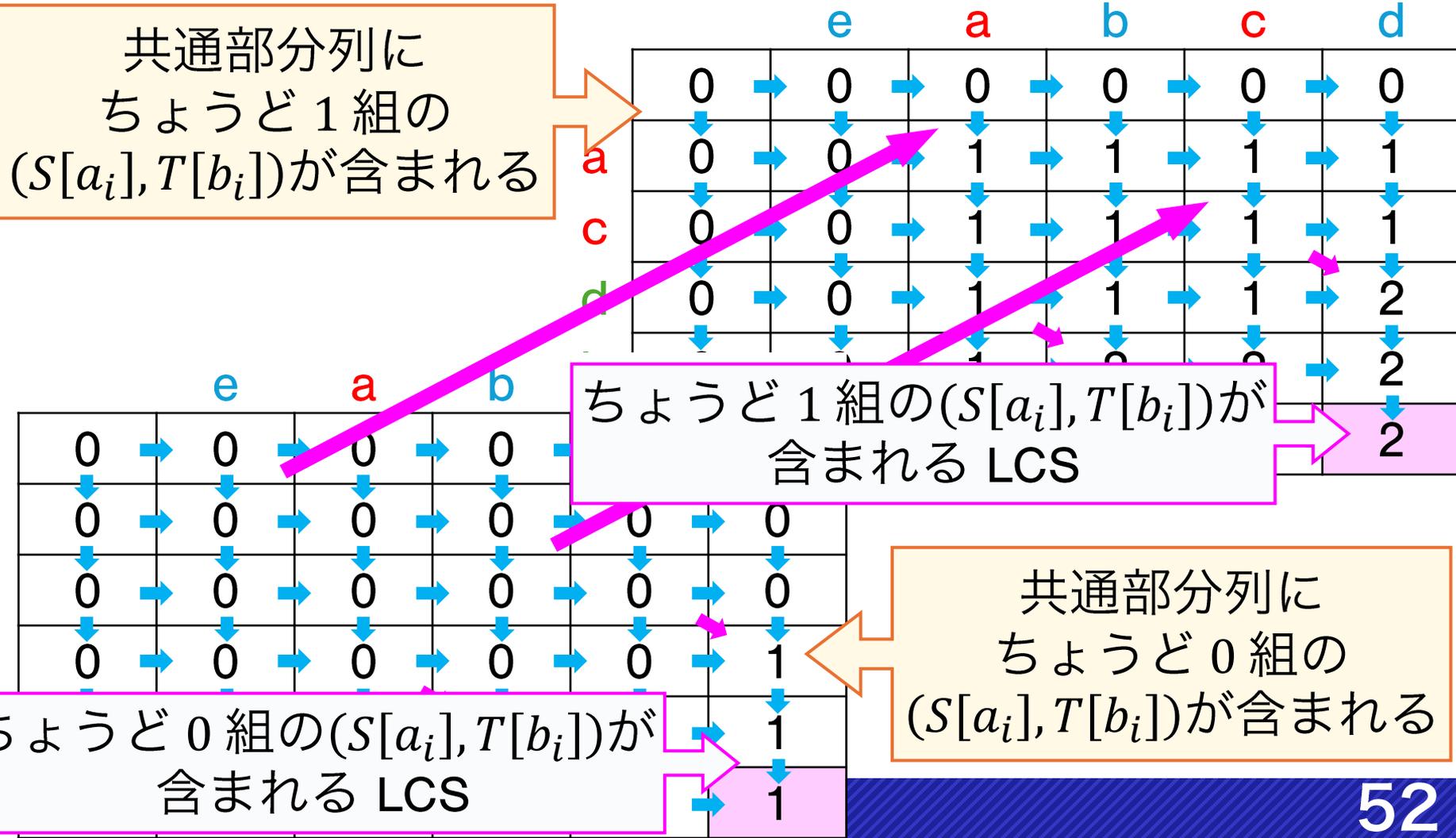
		e	a	b	c	d
	0	0	0	0	0	0
a	0	0	0	0	0	0
c	0	0	0	0	0	0
d	0	0	0	0	0	1
b	0	0	0	1	1	1
e	0	1	1	1	1	1

共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

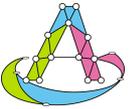
BD-LCS 問題に対するアルゴリズム



例 $S = e a b c d$, $T = a c d b e$, $Z = a c$, $A = (2, 4)$, $B = (1, 2)$



BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$ ちょうど $|Z|$ 組

ちょうど $|Z| - 1$ 組

ちょうど $|Z| - 2$ 組

ちょうど 0 組

共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

← $|Z|$ 組

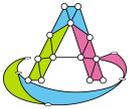
← $|Z| - 1$ 組

← $|Z| - 2$ 組

← 1 組

ちょうど 0 組の $(S[a_i], T[b_i])$ が
含まれる LCS

変更制約付き最長共通部分列問題 (再)

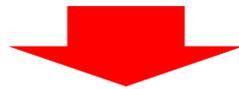


変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する
添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

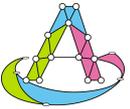
条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: ~~k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の
長さを求める~~



$(|Z| - k)$ 組み以上の $(S[a_i], T[b_i])$ からなる, LCS の
長さを求める

BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$ ちょうど $|Z|$ 組

ちょうど $|Z| - 1$ 組

ちょうど $|Z| - 2$ 組

ちょうど 0 組

共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

ちょうど $|Z|$ 組

ちょうど $|Z| - 1$ 組

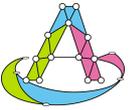
ちょうど $|Z| - 2$ 組

ちょうど $|Z| - k$ 組

ちょうど 1 組

ちょうど 0 組の $(S[a_i], T[b_i])$ が
含まれる LCS

BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$ ちょうど $|Z|$ 組

ちょうど $|Z| - 1$ 組

ちょうど $|Z| - 2$ 組

ちょうど 0 組

共通部分列に
ちょうど 0 組の
 $(S[a_i], T[b_i])$ が含まれる

$|Z|$ 組

$|Z| - 1$ 組

$|Z| - 2$ 組

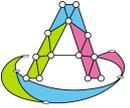
$|Z| - k$ 組

1 組

最大値が解

ちょうど 0 組の $(S[a_i], T[b_i])$ が
含まれる LCS

BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$ ちょうど $|Z|$ 組

ちょうど $|Z| - 1$ 組

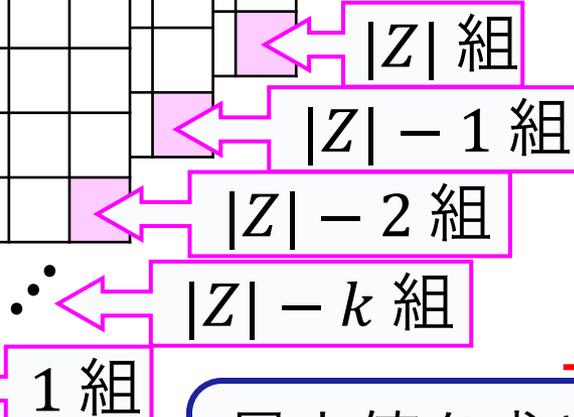
ちょうど $|Z| - 2$ 組

ちょうど 0 組

共通部分列に
ちょうど 0 組の

テーブルを埋める：
 $(|S| + 1)(|T| + 1)(|Z| + 1)$ ステップ

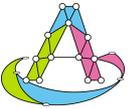
ちょうど 0 組の $(S[a_i], T[b_i])$ が
含まれる LCS



最大値が解

最大値を求める：
 $k + 1$ ステップ

BD-LCS 問題に対するアルゴリズム



$dp[i][j][\ell]$: (i, j, ℓ) に到達するまでに通る赤い矢印の最大値

$0 \leq i \leq |S|, 0 \leq j \leq |T|, 0 \leq \ell \leq |Z|$ ちょうど $|Z|$ 組

ちょうど $|Z| - 1$ 組

ちょうど $|Z| - 2$ 組

ちょうど 0 組

共通部分列に
ちょうど 0 組の

$O(|S||T||Z|)$

$|Z|$ 組

$|Z| - 1$ 組

$|Z| - 2$ 組

\vdots
 $|Z| - k$ 組

1 組

最大値が解

テーブルを埋める：
 $(|S| + 1)(|T| + 1)(|Z| + 1)$ ステップ

最大値を求める：
 $k + 1$ ステップ

ちょうど 0 組の $(S[a_i], T[b_i])$ が
含まれる LCS

まとめ



変更制約付き最長共通部分列問題 (BD-LCS 問題)

入力: S, T , 初期共通部分列 $Z = z_0 z_1 \dots z_{|Z|-1}$, S, T に対する

添字列 $A = (a_0, a_1, \dots, a_{|Z|-1})$, $B = (b_0, b_1, \dots, b_{|Z|-1})$, $k \in \mathbb{N}_0$

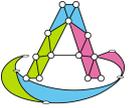
条件: $S[a_i] = T[b_i] = Z[i]$ ($0 \leq i \leq |Z| - 1$)

目的: k 組み以下の $(S[a_i], T[b_i])$ を削除するとき, LCS の長さを求める

定理

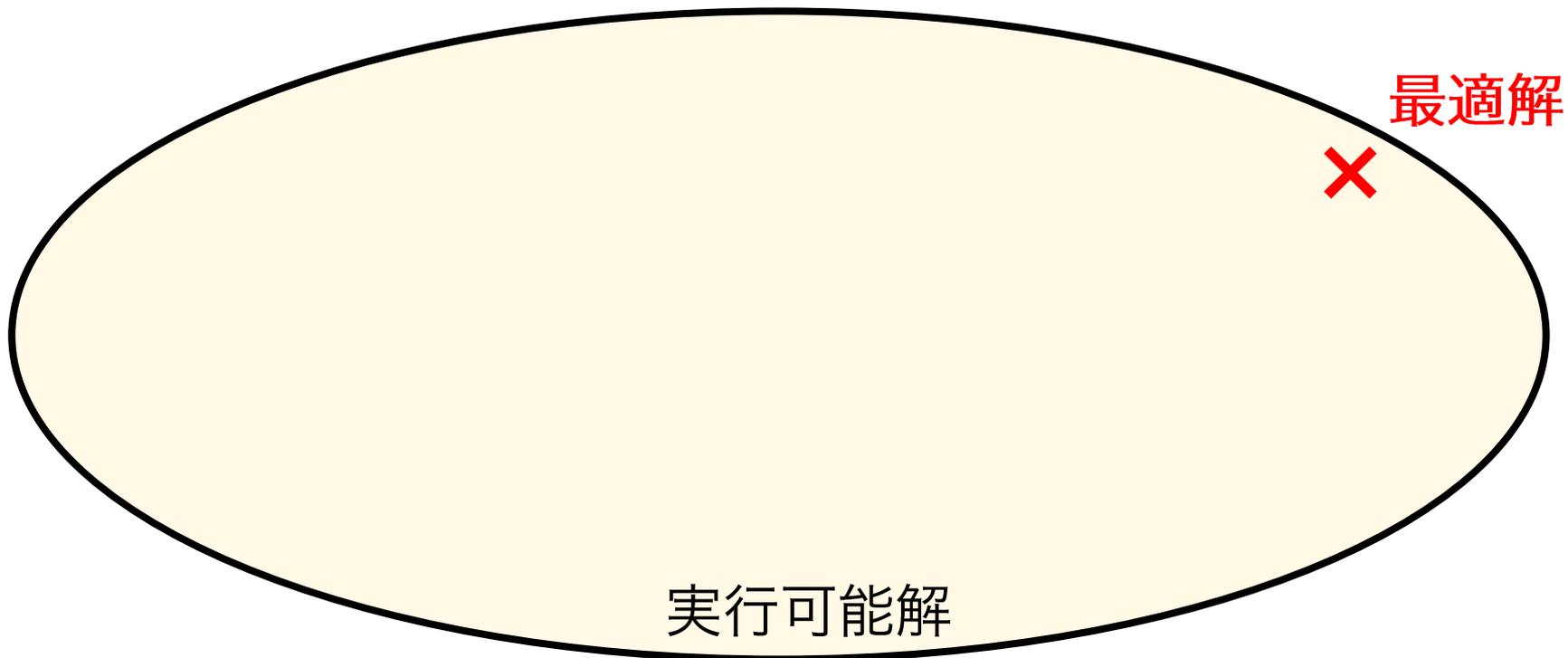
文字列 S, T と, 初期共通部分列 Z に対して $O(|S||T||Z|)$ の計算時間量で BD-LCS 問題の解を求めることができる。

本研究の背景



増分最適化 [Robert A et al., 1974]

最適化問題に対して初期解を与え, 変更を加えることで最適解を求める手法.



本研究の背景



増分最適化 [Robert A et al., 1974]

最適化問題に対して初期解を与え, 変更を加えることで最適解を求める手法.

