

課題 3

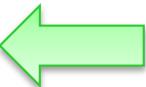
「文字列処理と動的計画法」

伊藤 康一, 吉留 崇

2018年度プログラミング演習A

本課題で学ぶこと

■ 編集距離

◆ 文字列処理  課題 1

◆ 再帰的手続き  課題2

◆ 動的計画法

- 問題3-0: 編集距離の解説
- 問題内にも解説あり

編集距離 (Edit Distance)

2つの文字列がどのくらい似ているかを表す指標

(例) 「マテオ」、「マンガ」

どちらが「マリオ」に似ている？

マリオ マリオ 「マテオ」の方が
マテオ マンガ 似ている

編集距離
(後で定義)

1

2

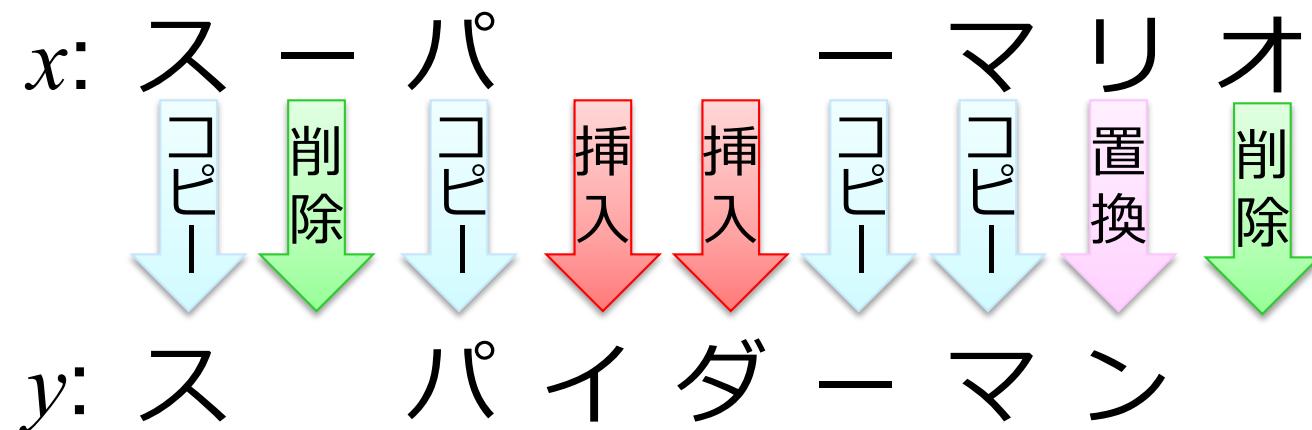
スペルチェッカーやDNAの配列の解析に応用

編集距離 (Edit Distance)

編集距離を求めるために、編集操作を行う

文字列 x にコピー、置換、挿入、削除を行い、
文字列 y に等しい文字列を得る操作

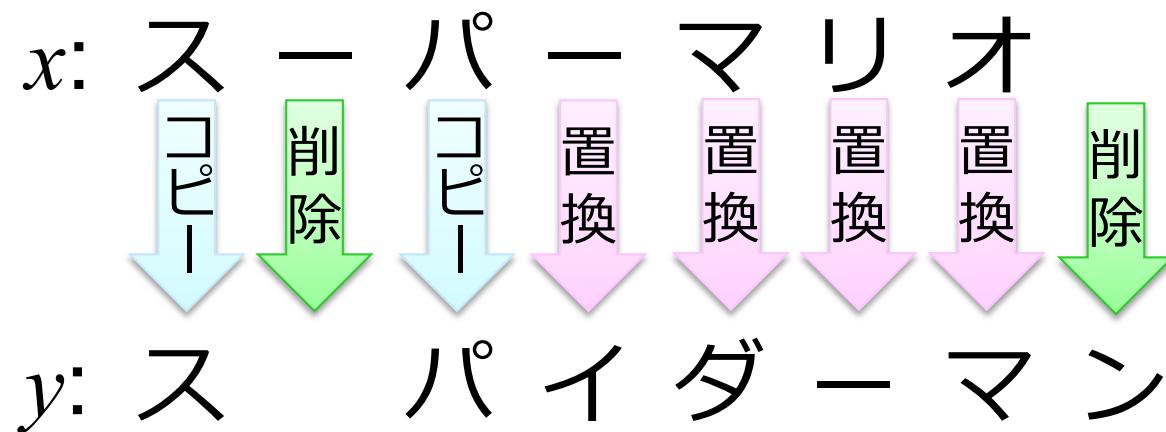
(例)「スーパーマリオ」と「スパイダーマン」



編集距離 (Edit Distance)

編集操作は、複数存在

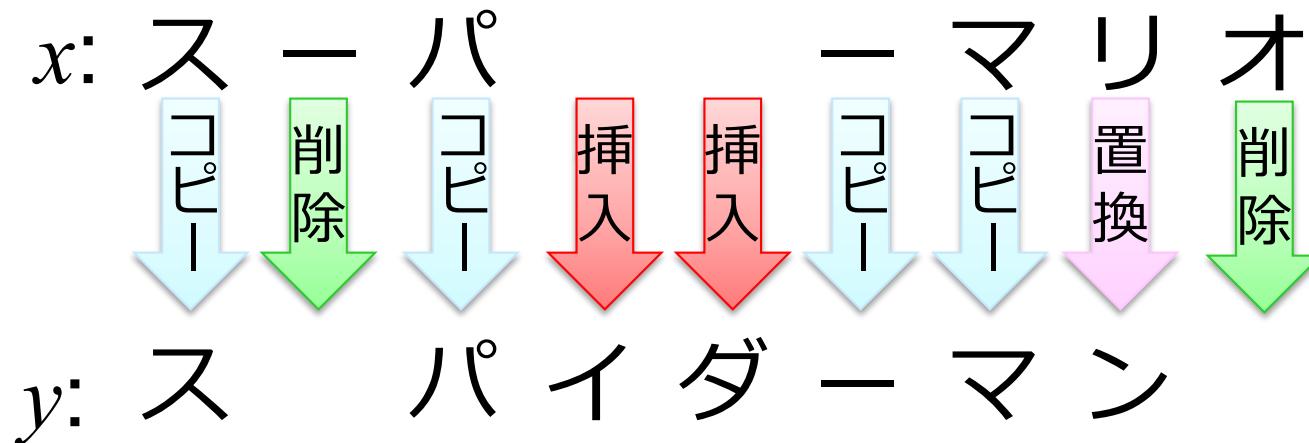
(例)「スーパーマリオ」と「スパイダーマン」



編集距離 (Edit Distance)

文字列 x を文字列 y に変換する時の「削除」，「挿入」，
「置換」の最小回数（「コピー」はカウントしない）

(例)「スーパーマリオ」と「スパイダーマン」



編集距離 = 5

編集距離の計算方法

文字列に対する数学記号

	1	2	3	4	5	6	7	8	9
x :	a	b	b	a	b	a	a	a	b

文字列 x の i 番目の文字 : x_i 例) $x_4 = a$

文字列 x の先頭から i 番目までの文字列 : X_i  接頭辞

例) $X_4 = abba$

長さ 0 の文字列 : ε

編集距離の計算方法

文字列 $x = \langle x_1, x_2, \dots, x_m \rangle$, $y = \langle y_1, y_2, \dots, y_n \rangle$ の
接頭辞 X_i , Y_j の編集距離: $c_{i,j}$

$$c_{i,j} = \begin{cases} \max(i, j) & (i = 0 \text{ または } j = 0 \text{ の時}) \\ \min(c_{i-1, j-1} + d(x_i, y_j), \\ \quad c_{i-1, j} + 1, \\ \quad c_{i, j-1} + 1) & (\text{その他}) \end{cases}$$

$$d(x_i, y_j) = \begin{cases} 1 & (x_i \neq y_j) \\ 0 & (x_i = y_j) \end{cases}$$

* クロネッカーのデルタ
ではない事に注意

編集距離の計算方法

$$c_{i,j} = \min(c_{i-1,j-1} + d(x_i, y_j), \begin{cases} 1 & (x_i \neq y_j) \\ 0 & (x_i = y_j) \end{cases})$$

削除 置換
コピ- 挿入

(例) $x=ab$ 、 $y=cd$

$c_{2,2}<ab, cd>$ の編集距離は、以下の距離の最小値

$$c_{1,1}<a, c> + d(b, d)$$

a b
↓ 置換
c d

$$c_{1,2}<a, cd> + 1$$

a b
↓ 削除
c d

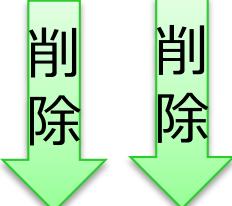
$$c_{2,1}<ab, c> + 1$$

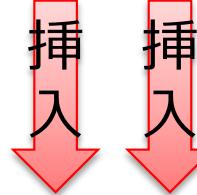
a b
↓ 挿入
c d ,

編集距離の計算方法

$$c_{i,j} = \max(i, j) \quad (i = 0 \text{または} j = 0 \text{の時})$$

$$c_{i,0} = i, \text{ 削除} \quad c_{0,j} = j \text{ 挿入}$$

$x :$ a b
y :
 削除

$x :$
 挿入
 $y :$ c d

編集距離の計算方法

$$c_{i,j} = \min(c_{i-1,j-1} + d(x_i, y_j), \begin{cases} 1 & (x_i \neq y_j) \\ 0 & (x_i = y_j) \end{cases})$$

削除 挿入

$$c_{i-1,j} + 1, \quad c_{i,j-1} + 1)$$

$$c_{i,0} = i, \quad \text{削除} \quad \quad c_{0,j} = j \quad \text{挿入}$$

再帰関数による実装

動的計画法による実装

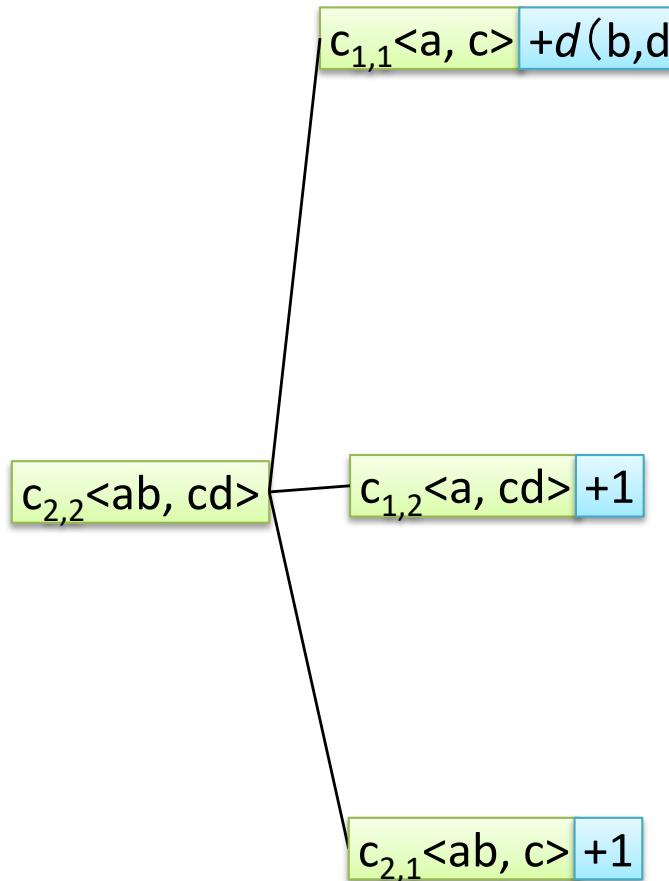
再帰的に計算

c_{2,2}<ab, cd>

$$c_{i,j} < X_i, Y_j > = \min(c_{i-1,j-1} + \delta(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} < X_i, \varepsilon > = i, \quad c_{0,j} < \varepsilon, Y_j > = j$$

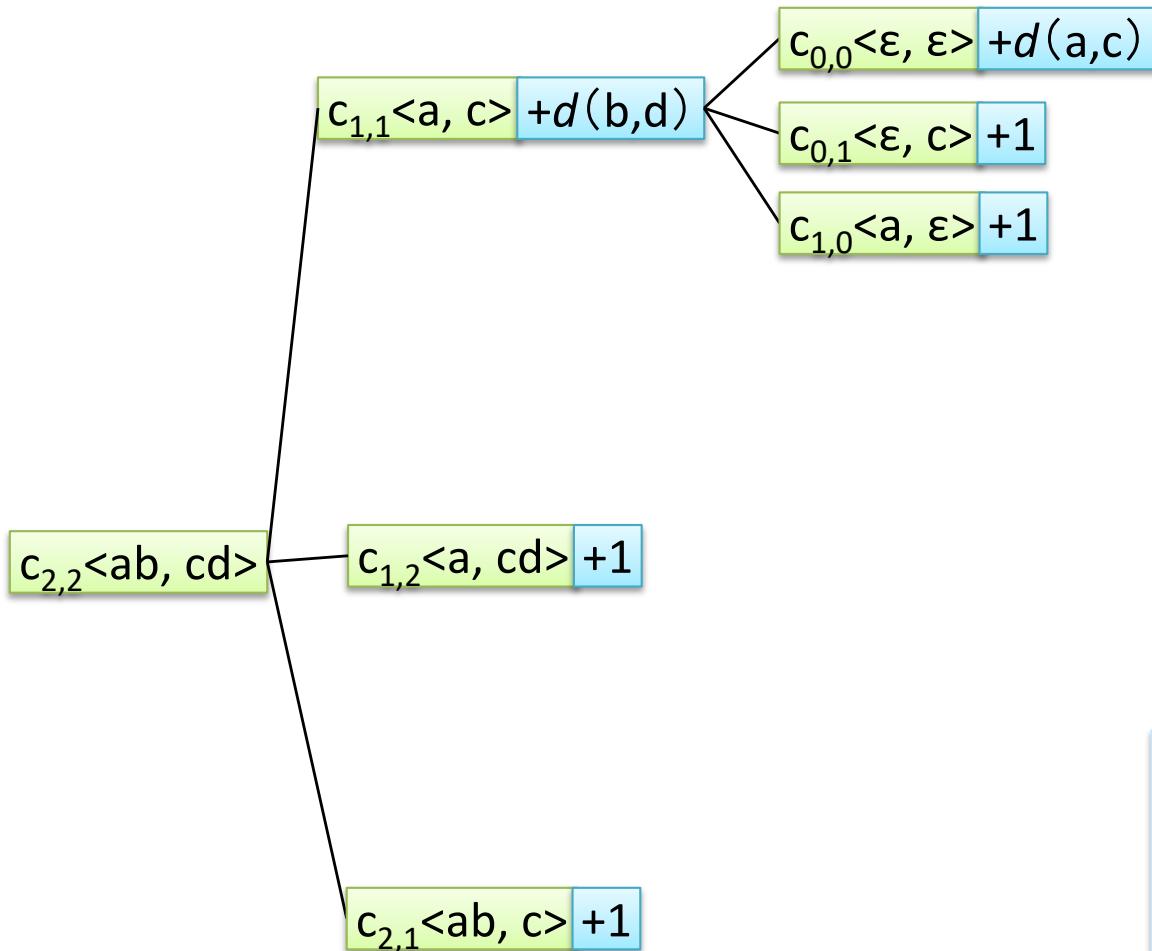
再帰的に計算



$$c_{i,j} < X_i, Y_j > = \min(c_{i-1,j-1} + d(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} < X_i, \varepsilon > = i, \quad c_{0,j} < \varepsilon, Y_j > = j$$

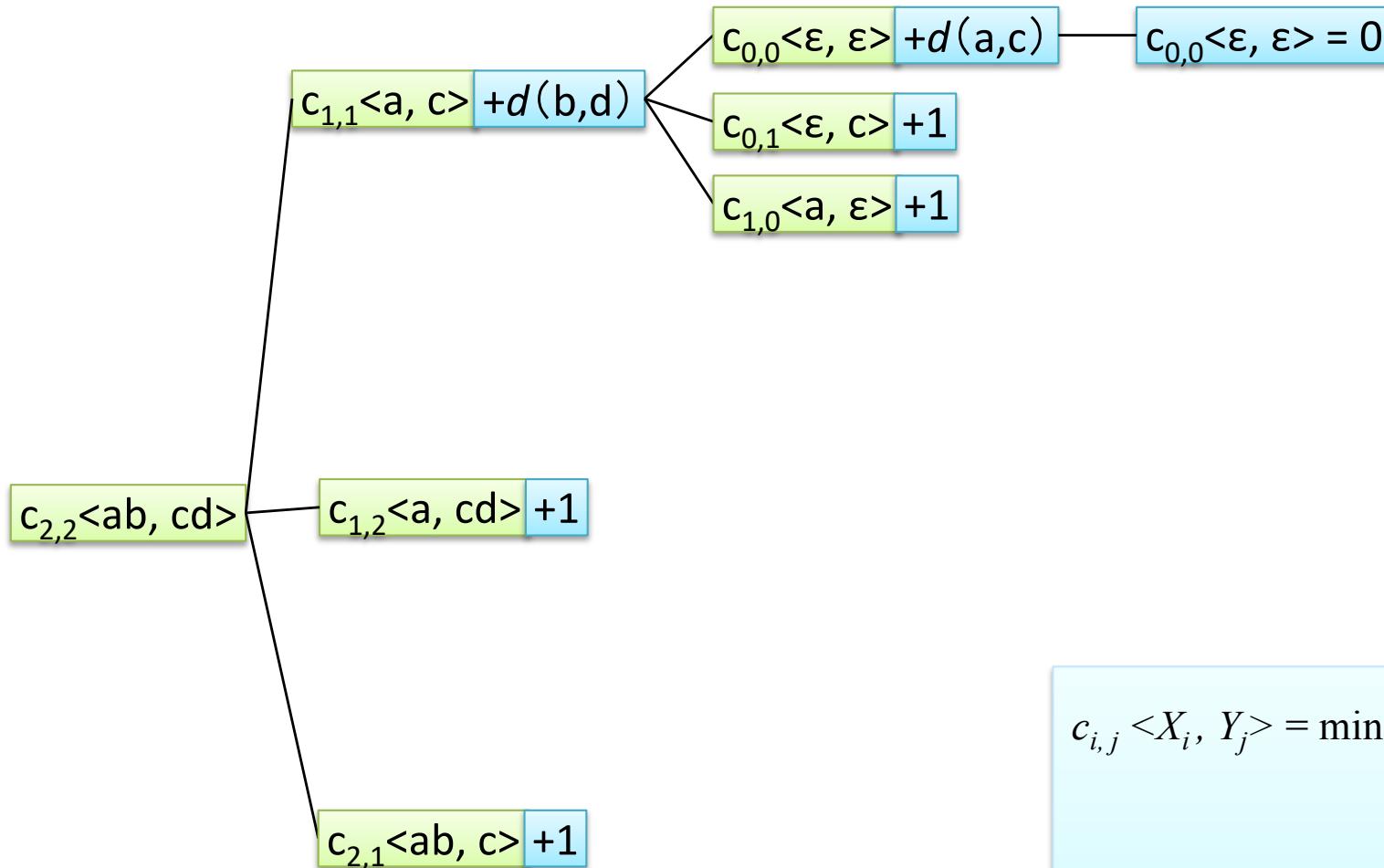
再帰的に計算



$$c_{i,j} < X_i, Y_j > = \min(c_{i-1,j-1} + d(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} < X_i, \varepsilon > = i, \quad c_{0,j} < \varepsilon, Y_j > = j$$

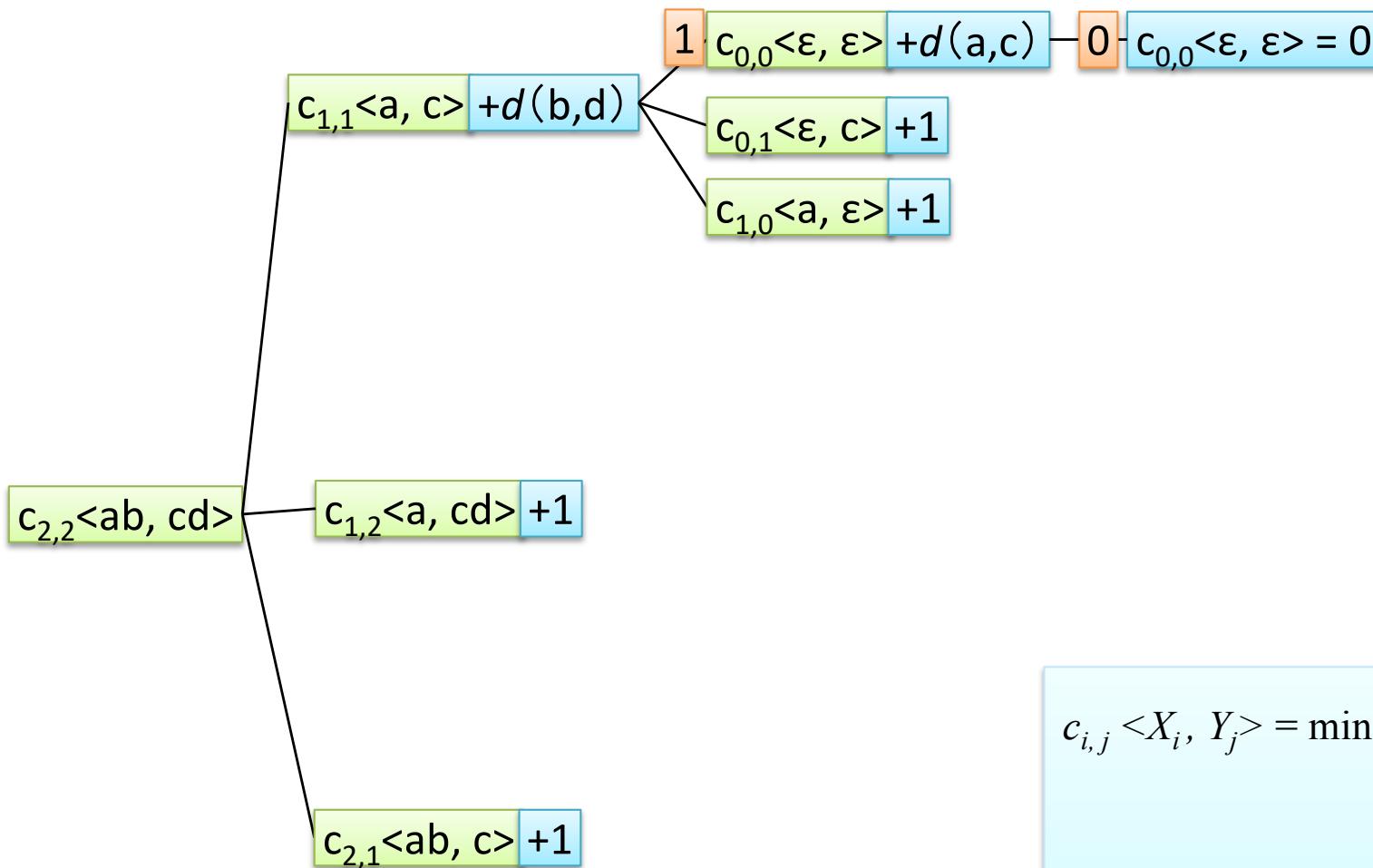
再帰的に計算



$$c_{i,j} < X_i, Y_j > = \min(c_{i-1,j-1} + d(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} < X_i, \varepsilon > = i, \quad c_{0,j} < \varepsilon, Y_j > = j$$

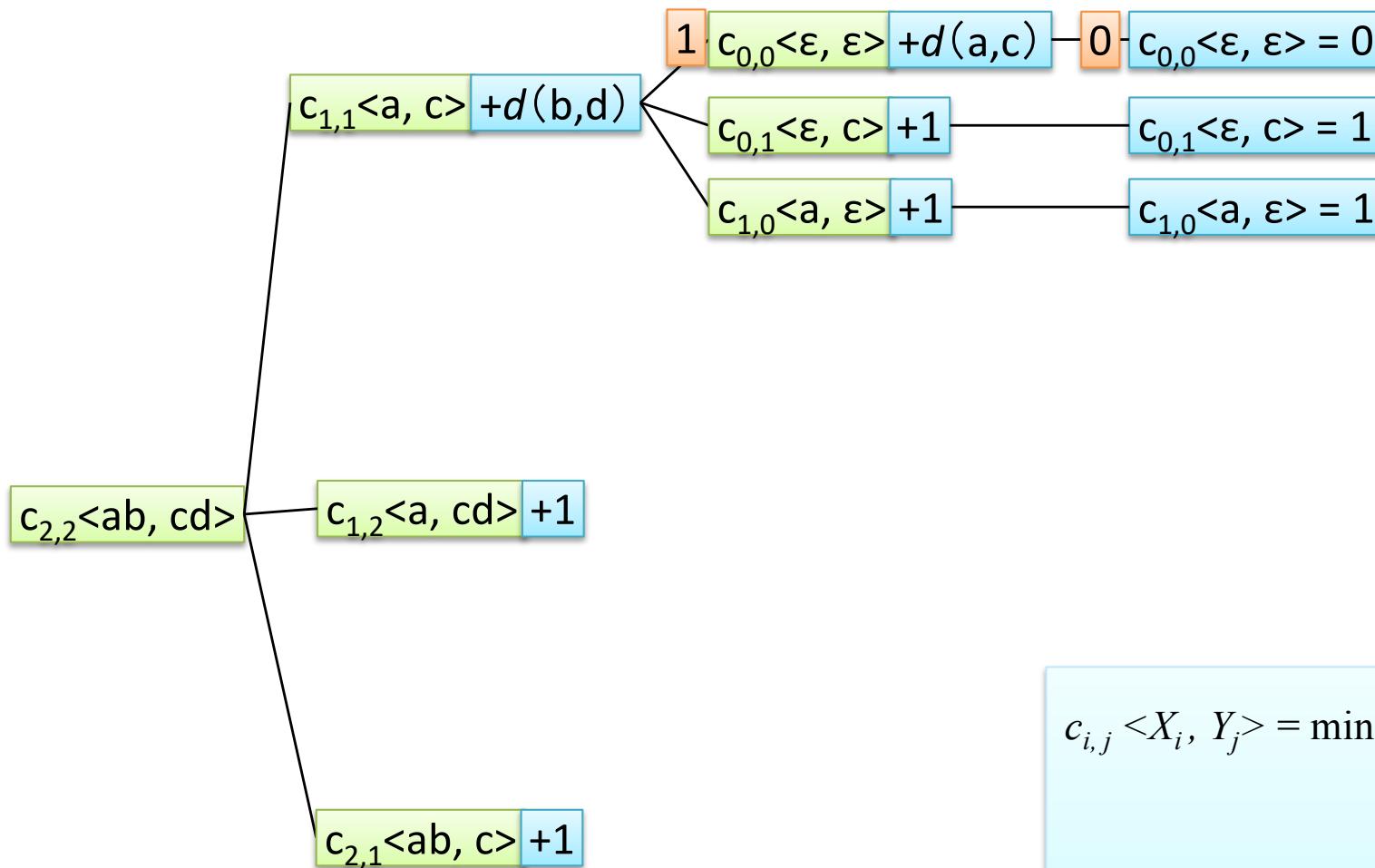
再帰的に計算



$$c_{i,j} \langle X_i, Y_j \rangle = \min(c_{i-1,j-1} + d(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} \langle X_i, \varepsilon \rangle = i, \quad c_{0,j} \langle \varepsilon, Y_j \rangle = j$$

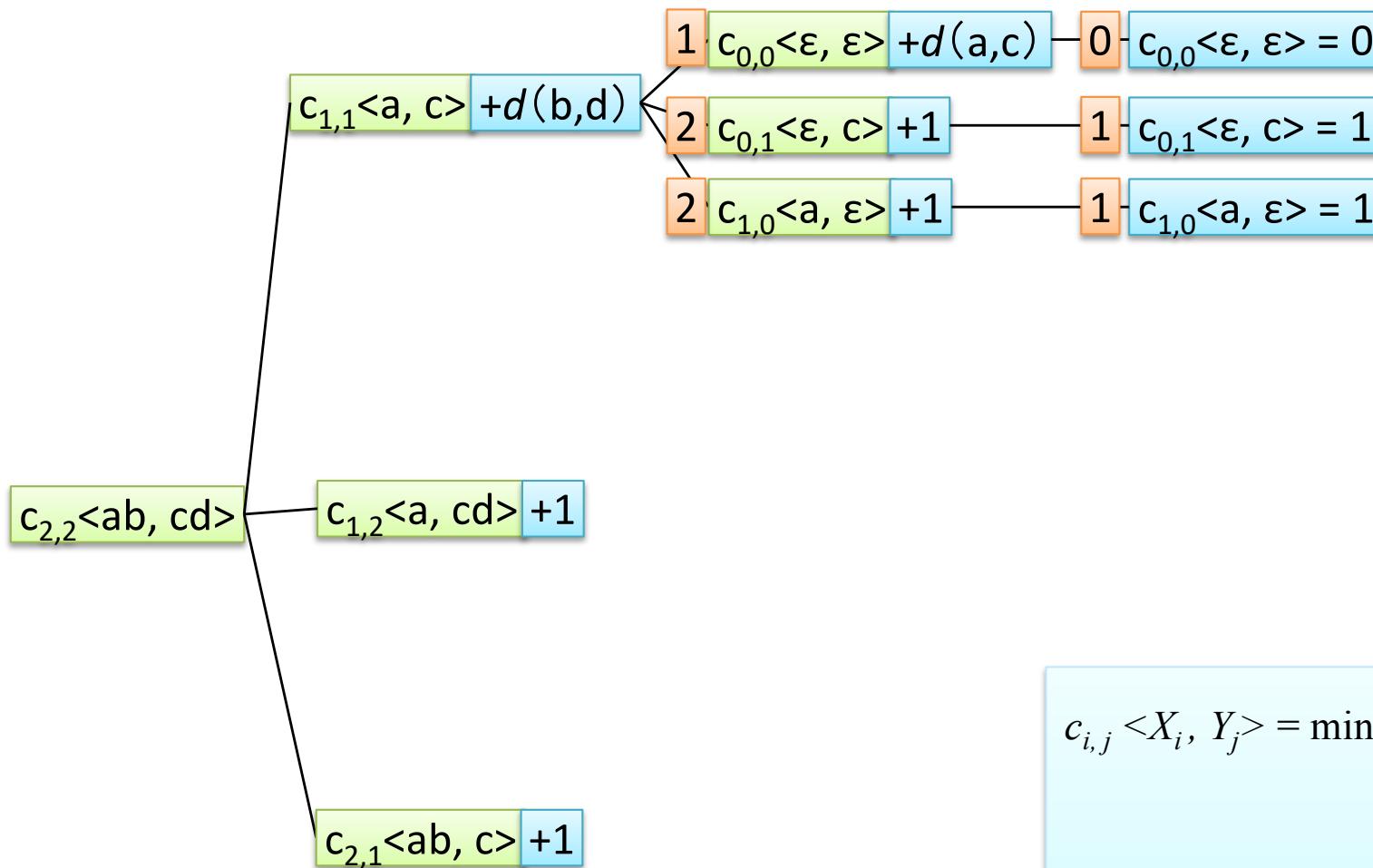
再帰的に計算



$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + d(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \epsilon> = i, \quad c_{0,j} <\epsilon, Y_j> = j$$

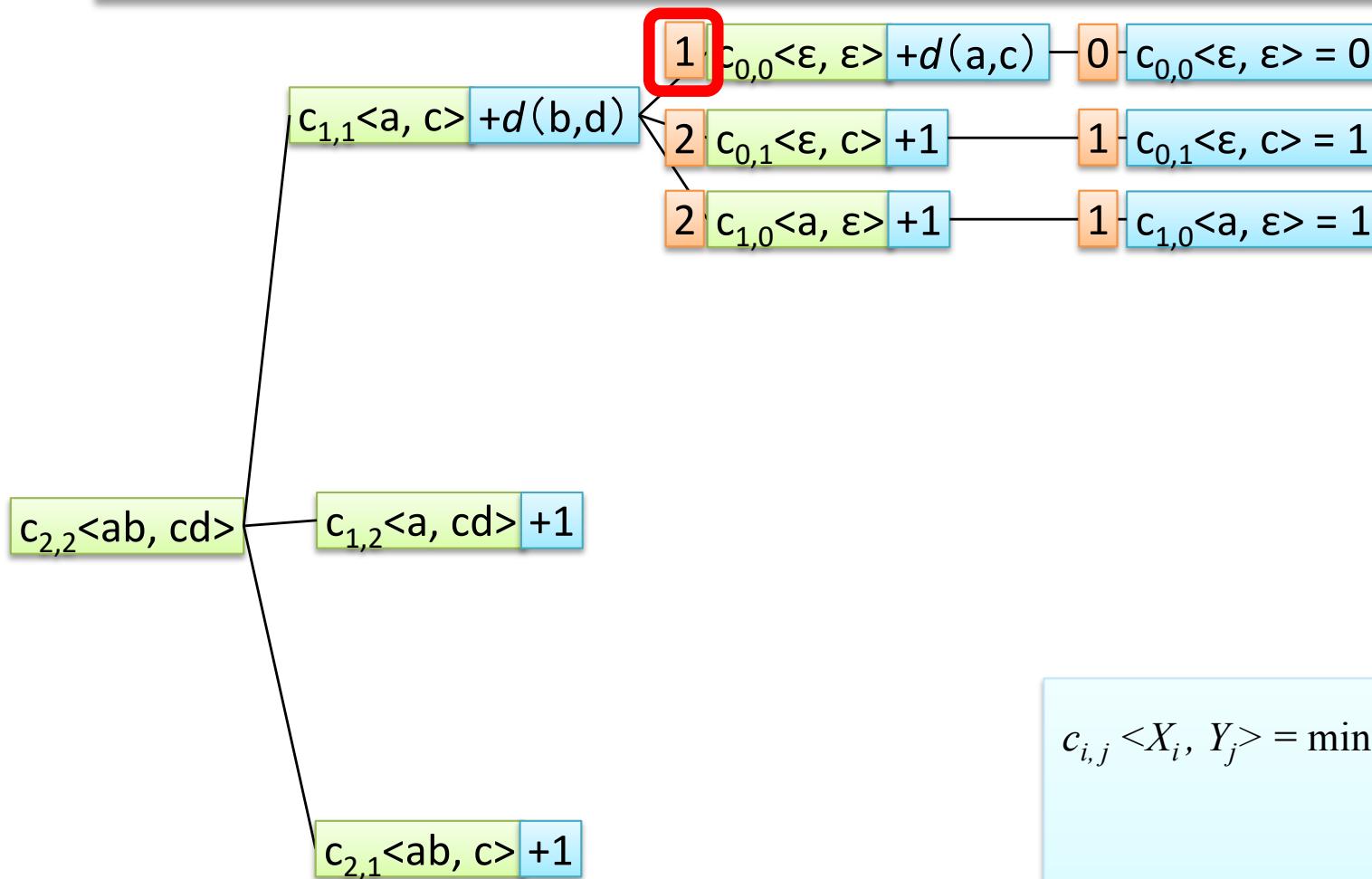
再帰的に計算



$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + d(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \epsilon> = i, \quad c_{0,j} <\epsilon, Y_j> = j$$

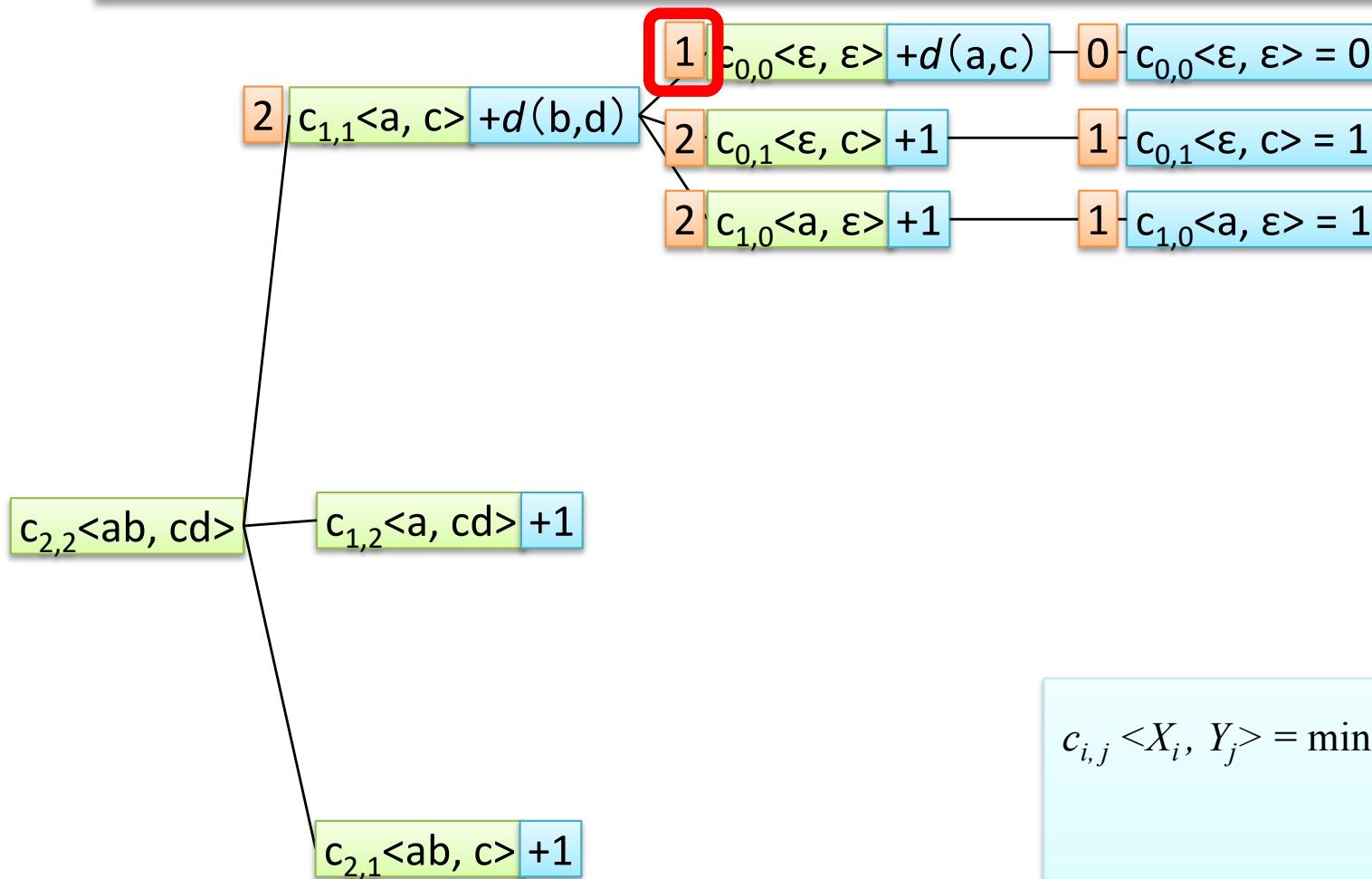
再帰的に計算



$$c_{i,j} < X_i, Y_j > = \min(c_{i-1,j-1} + d(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} < X_i, \varepsilon > = i, \quad c_{0,j} < \varepsilon, Y_j > = j$$

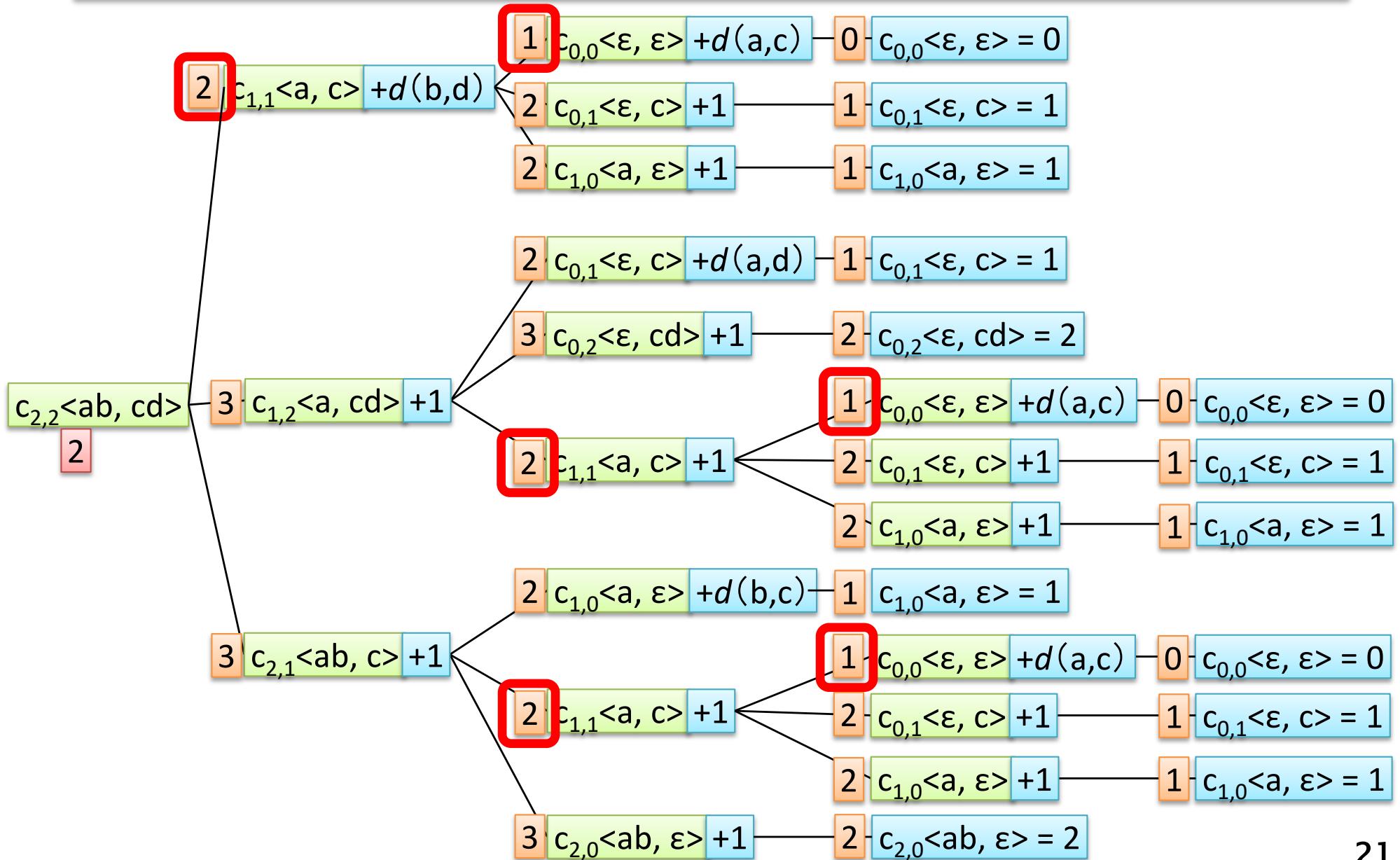
再帰的に計算



$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + d(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \epsilon> = i, \quad c_{0,j} <\epsilon, Y_j> = j$$

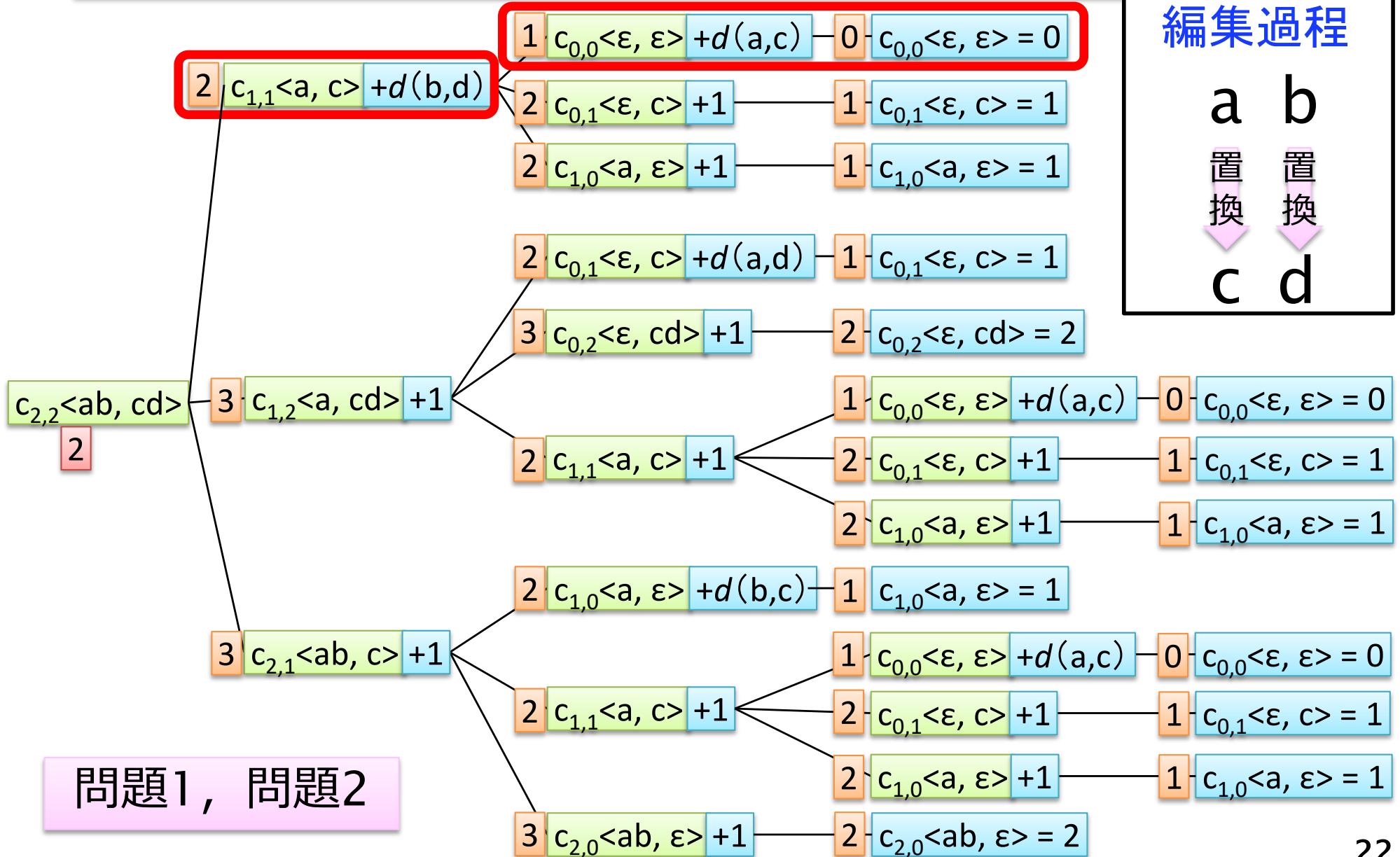
再帰的に計算



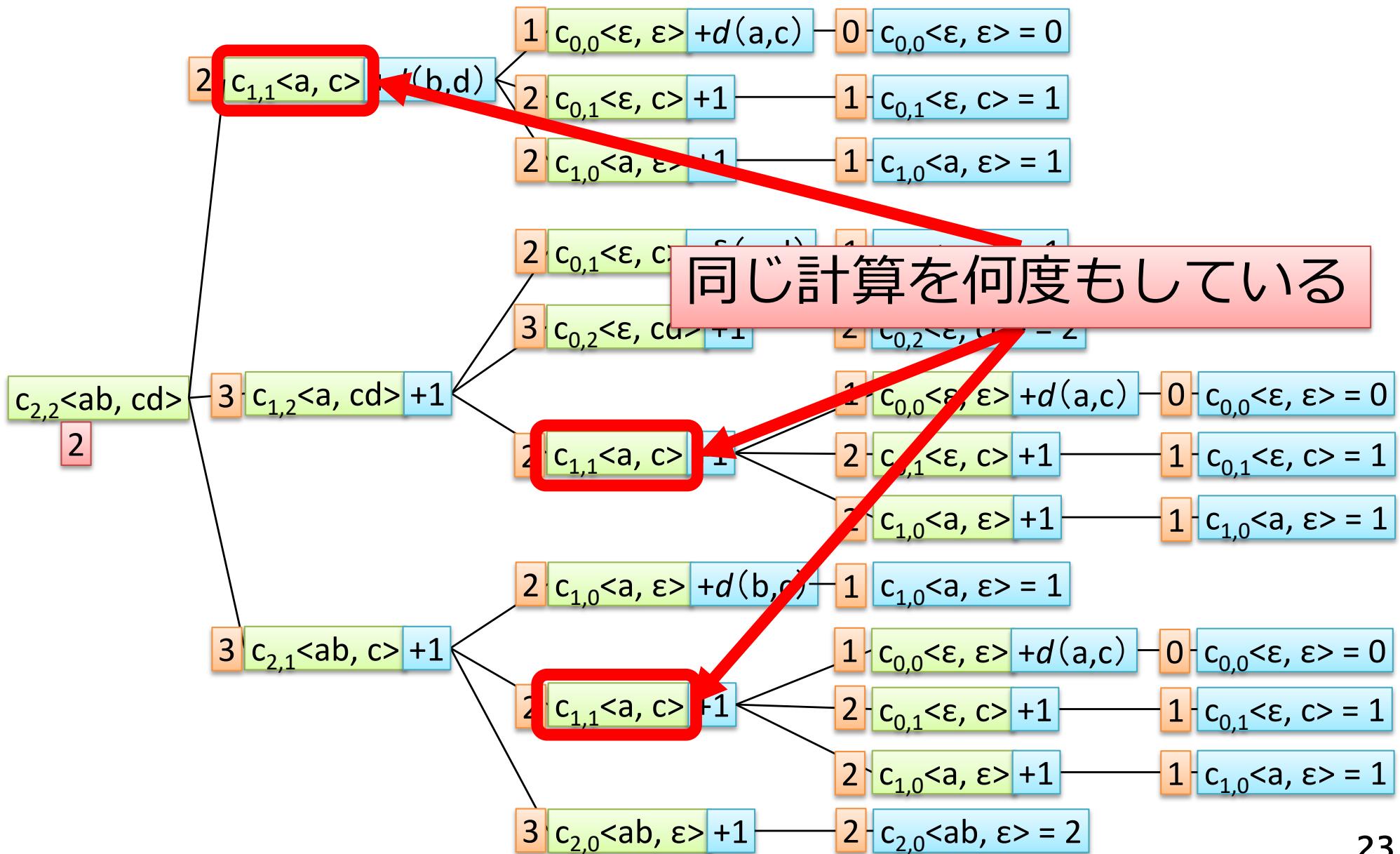
再帰的に計算

編集距離を
与える
編集過程

a b
置換 置換
↓ ↓
c d



再帰的計算の問題点



再帰的計算の問題点

- 同じ計算を何度も行うため計算量大



メモ化 (memoization) による対策

一度計算したものをメモとして記憶, 次に計算するときはそのメモを呼び出す

問題5

動的計画法(Dynamic Programming, DP)

$c_{i,j} <ab, cd>$

	ε	c	d
ε	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$
a	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$
b	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$C_{0,0}, C_{0,1}, C_{1,0}$ が計算済み
ならば、 $C_{1,1}$ の値分かる

動的計画法(Dynamic Programming, DP)

$c_{i,j} <ab, cd>$

	ε	c	d
ε	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$
a	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$
b	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$C_{1,1}, C_{1,2}, C_{2,1}$ が計算済み
ならば、 $C_{2,2}$ の値分かる

動的計画法(Dynamic Programming, DP)

$c_{i,j} <ab, cd>$

	ε	c	d
ε	$C_{0,0}$	$C_{0,1}$	$C_{0,2}$
a	$C_{1,0}$	$C_{1,1}$	$C_{1,2}$
b	$C_{2,0}$	$C_{2,1}$	$C_{2,2}$

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

ボトムアップに $C_{i,j}$ を求める



編集距離 $C_{2,2}$ 得られる

同じ計算は1度のみ \Rightarrow 計算量減

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε			
a			
b			

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0  1  2		
a			
b			

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$$c_{0,0} <\varepsilon, \varepsilon> = 0, c_{0,1} <\varepsilon, c> = 1, c_{0,2} <\varepsilon, cd> = 2$$

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0	1	2
a	1		
b	2		

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$$c_{0,0} <\varepsilon, \varepsilon> = 0, c_{1,0} <a, \varepsilon> = 1, c_{2,0} <ab, \varepsilon> = 2$$

動的計画法による計算

$c_{i,j} <ab, cd>$

		0+ $\delta(a,c)$ = 0+1=1	C	d
ε	0	1	1+1=2	
a	1	1		
b	2			

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$$c_{1,1} <a, c> = \min(c_{0,0} <\varepsilon, \varepsilon> + \delta(a, c), c_{0,1} <\varepsilon, c> + 1, c_{1,0} <a, \varepsilon> + 1)$$

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε			d
ε	0	1	2	
a	1	1	2	
b	2			

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$2+1=3 <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$$c_{1,2} <a, cd> = \min(c_{0,1} <\varepsilon, c> + \delta(a, d), c_{0,2} <\varepsilon, cd> + 1, c_{1,1} <a, c> + 1)$$

動的計画法による計算

$$c_{i,j} <ab, cd>$$

	ε	c	d
ε	$1 + \delta(b, c) = 1 + 1 = 2$	1	2
a	1	1	$1 + 1 = 2$
b	2	2	$2 + 1 = 3$

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), \\ c_{i-1,j} + 1, \\ c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$$c_{2,1} <ab, c> = \min(c_{1,0} <a, \varepsilon> + \delta(b, c), \\ c_{1,1} <a, c> + 1, \\ c_{2,0} <ab, \varepsilon> + 1)$$

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0	$1 + \delta(b, d) = 1 + 1 = 2$	2
a	1	1	2
b	2	2	2

Diagram illustrating the computation of $c_{i,j} <ab, cd>$ using dynamic programming. The table shows the cost of matching substrings. A red circle highlights 'd' in the top row, and another red circle highlights 'b' in the bottom row. Orange callouts show calculations: $1 + \delta(b, d) = 1 + 1 = 2$ for the cell (b, d), and $2 + 1 = 3$ for the cells (a, d) and (b, a).

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$

$$c_{2,2} <ab, cd> = \min(c_{1,1} <a, c> + \delta(b, d), c_{1,2} <a, cd> + 1, c_{2,1} <ab, c> + 1)$$

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0	1	2
a	1	1	2
b	2	2	2

$$c_{i,j} <X_i, Y_j> = \min(c_{i-1,j-1} + \delta(x_i, y_j), c_{i-1,j} + 1, c_{i,j-1} + 1)$$

$$c_{i,0} <X_i, \varepsilon> = i, c_{0,j} <\varepsilon, Y_j> = j$$



ここが編集距離

問題3, 問題4

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0	1	2
a	1	1	2
b	2	2	2

編集操作の求め方

値を埋める際に使ったデータ
の流れの矢印を表示

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0 ← ↑	1 ← ↑	2
a	1 ↑	1 ↑	2 ← ↑
b	2	2	2

編集操作の求め方

矢印を180°回転

動的計画法による計算

$c_{i,j} <ab, cd>$

	ε	c	d
ε	0	1	2
a	1	1	2
b	2	2	2

編集操作の求め方

右下から0の部分までたどって
いったときの矢印が編集操作
(ただし、編集操作が1つに決
まるとは限らない)

a b
置換 置換
c d

問題6

R R

課題3：全部で7問

- 問題1 (10点) [記述問題]再帰による編集距離の計算 (手書き可)
- 問題2 (10点) 再帰関数による実装
- 問題3 (10点) [記述問題]動的計画法による編集距離の計算 (手書き可)
- 問題4 (10点) 動的計画法による実装
- 問題5 (20点) [プログラム+記述問題]再帰呼び出しにおけるメモ化
- 問題6 (10点) 編集操作の表示
- 問題7 (10点) スペルチェッカーの作成

2週目の面接
のみ受付

面接を受けるときの注意事項

- C言語の用語や関数の動作をあらかじめ調べてくること
プログラムのソースコードに説明を書いてかまわないので,面接時に説明できるようにすること
- 課題に関する問題文をきちんと読むこと
問題文を読むことが解答への近道になることもあります
- 2週目の面接時に,途中だったとしても,取り組んでいる問題を持ってきましょう
困っていることやわからないことがあれば, 面接の時にアドバイスをするので ; 途中のプログラムでも遠慮なく持ってきてましょう