

データ構造と アルゴリズムII

第4回
動的計画法 3

179

演習: ロッド切り出し問題

- 長さ10mの金属棒がある. これを切り出して販売する(切り出す長さは1mの倍数). 長さ i の加工された金属棒の販売価格 p_i は以下で与えられる. 販売価格の合計を最大にする切り出し方法を求めよ.

length i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	22	23

$$\max(p_i, \max_{1 \leq j \leq \lfloor \frac{i}{2} \rfloor} (v_j + v_{i-j}))$$

180

演習: ロッド切り出し問題

length i	1	2	3	4	5	6	7	8	9	10
p_i	1	5	8	9	10	17	17	20	22	23
v_i :	1	5	8	10	13	17	18	22	25	27

$$\begin{matrix} 10 \\ 4-6 \\ 2-2-6 \end{matrix} \max(p_i, \max_{1 \leq j \leq \lfloor \frac{i}{2} \rfloor} (v_j + v_{i-j}))$$

181

演習: ロッド切り出し問題

- ロッド切り出し問題で, 切る回数が最大3等の制限を加えると, 部分問題最適性が保証されなくなる
- この課題をクリアして, 動的計画法を使うにはどうしたらよいか?

182

例題 最長共通部分列

183

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

184

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

– 部分系列とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ の部分系列が $Z=\langle z_1, z_2, \dots, z_k \rangle$ であるとは、増加する添え字の列 $\langle i_1, i_2, \dots, i_k \rangle$ が存在し、 $x_{i_j}=z_j$ を満たす。

$X=\langle A, B, C, B, D, A, B \rangle$
 $Z=\langle B, C, D, B \rangle$

部分系列

185

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

– 部分系列とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ の部分系列が $Z=\langle z_1, z_2, \dots, z_k \rangle$ であるとは、増加する添え字の列 $\langle i_1, i_2, \dots, i_k \rangle$ が存在し、 $x_{i_j}=z_j$ を満たす。

$X=\langle A, \overset{\langle 2, 3, 5, 7 \rangle}{B, C, B, D, A, B}$
 $Z=\langle B, C, D, B \rangle$

部分系列

186

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

– 部分系列とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ の部分系列が $Z=\langle z_1, z_2, \dots, z_k \rangle$ であるとは、増加する添え字の列 $\langle i_1, i_2, \dots, i_k \rangle$ が存在し、 $x_{i_j}=z_j$ を満たす。

– 共通部分系列とは
 2つの系列の共通する部分系列

$X=\langle A, B, C, B, D, A, B \rangle$
 $Y=\langle B, D, C, A, B, A \rangle$

$\langle B, C, A \rangle$ は
 共通部分系列

187

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

– 部分系列とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ の部分系列が $Z=\langle z_1, z_2, \dots, z_k \rangle$ であるとは、増加する添え字の列 $\langle i_1, i_2, \dots, i_k \rangle$ が存在し、 $x_{i_j}=z_j$ を満たす。

– 共通部分系列とは
 2つの系列の共通する部分系列

$X=\langle A, \overset{\circ}{B}, \overset{\circ}{C}, B, D, \overset{\circ}{A}, B \rangle$
 $Y=\langle \overset{\circ}{B}, D, \overset{\circ}{C}, \overset{\circ}{A}, B, A \rangle$

$\langle B, C, A \rangle$ は
 共通部分系列

188

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

– 部分系列とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ の部分系列が $Z=\langle z_1, z_2, \dots, z_k \rangle$ であるとは、増加する添え字の列 $\langle i_1, i_2, \dots, i_k \rangle$ が存在し、 $x_{i_j}=z_j$ を満たす。

– 共通部分系列とは
 2つの系列の共通する部分系列

– 最長共通部分系列問題とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ と $Y=\langle y_1, y_2, \dots, y_n \rangle$ の最長共通部分系列(LCS)を求める

$X=\langle A, \overset{\circ}{B}, \overset{\circ}{C}, B, D, \overset{\circ}{A}, \overset{\circ}{B} \rangle$
 $Y=\langle \overset{\circ}{B}, D, \overset{\circ}{C}, \overset{\circ}{A}, \overset{\circ}{B}, A \rangle$

最長共通部分系列(LCS)は
 $\langle B, C, A, B \rangle$ と $\langle B, D, A, B \rangle$

189

問題

- DNAの塩基がどれぐらい一致しているか知りたい
- 最長部分系列を求める

– 部分系列とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ の部分系列が $Z=\langle z_1, z_2, \dots, z_k \rangle$ であるとは、増加する添え字の列 $\langle i_1, i_2, \dots, i_k \rangle$ が存在し、 $x_{i_j}=z_j$ を満たす。

– 共通部分系列とは
 2つの系列の共通する部分系列

– 最長共通部分系列問題とは
 $X=\langle x_1, x_2, \dots, x_m \rangle$ と $Y=\langle y_1, y_2, \dots, y_n \rangle$ の最長共通部分系列(LCS)を求める

190

総当り作戦の破綻

- X の部分系列の個数
→ $O(2^m)$

総当りには $\Omega(2^m)$ 必要

191

アルゴリズムのステップ1: 最適解の構造を 特徴づける

192

LCSの部分構造最適性

- $X = \langle x_1, x_2, \dots, x_m \rangle$ と $Y = \langle y_1, y_2, \dots, y_n \rangle$ のLCSが $Z = \langle z_1, z_2, \dots, z_k \rangle$ であると仮定すると
1. $x_m = y_n$ ならば $z_k = x_m = y_n$ であり, Z_{k-1} は X_{m-1} と Y_{n-1} のLCSである
 2. $x_m \neq y_n$ のとき, $z_k \neq x_m$ ならば Z は X_{m-1} と Y のLCSである
 3. $x_m \neq y_n$ のとき, $z_k \neq y_n$ ならば Z は X と Y_{n-1} のLCSである

X_i は X の最初の i 個の文字からなる文字列
 $X_i = \langle x_1, x_2, \dots, x_i \rangle$ とする (接頭語)

193

LCSを求める手順

- $x_m = y_n$ のとき
– X_{m-1} と Y_{n-1} のLCSを求める
- $x_m \neq y_n$ のとき
– 「 X_{m-1} と Y のLCS」と「 X と Y_{n-1} のLCS」の長いほうを採用する

部分問題の部分問題を共有している
→ 部分問題重複性

194

最適解の値

$c[i, j]$ = X_i と Y_j のLCSの長さ

$$c[i, j] = \begin{cases} 0 & (i=0 \text{ または } j=0 \text{ のとき}) \\ c[i-1, j-1] + 1 & (i, j > 0 \text{ かつ } x_i = y_j \text{ のとき}) \\ \max(c[i, j-1], c[i-1, j]) & (i, j > 0 \text{ かつ } x_i \neq y_j \text{ のとき}) \end{cases}$$

※問題に依存する条件により, 検討すべき部分問題が変化する

195

アルゴリズムのステップ3: (ボトムアップに) 最適解の値を求める

196

動的計画法の利用

- 再帰(メモ化なし)で計算すると指数時間
- 部分問題の個数は $\Theta(nm)$ 個のみ

→ 動的計画法で
ボトムアップ(昇順)に解く

197

LCS-LENGTH(X,Y)

```

m ← length[X]
n ← length[Y]
for i ← 1 to m
  do c[i,0] ← 0
for j ← 1 to n
  do c[0,j] ← 0
for i ← 1 to m
  do for j ← 1 to n
    do if xi = yj
      then c[i,j] ← c[i-1,j-1]+1
         b[i,j] ← "左上"
    else if c[i-1,j] ≥ c[i,j-1]
      then c[i,j] ← c[i-1,j]
         b[i,j] ← "↑"
    else c[i,j] ← c[i,j-1]
         b[i,j] ← "←"

```

} i=0, j=0を設定

} 左上から1つずつ計算

198

例 (p33)

	j	0	1	2	3	4	5	6
i	y _j	B	D	C	A	B	A	
0	x _i	0	0	0	0	0	0	0
1	A	0						
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

i=0, j=0 を初期化

199

例 (p33)

	j	0	1	2	3	4	5	6
i	y _j	B	D	C	A	B	A	
0	x _i	0	0	0	0	0	0	0
1	A	0	↑0					
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

i=1, j=1 のとき
x_i≠y_j, c[i-1,j] ≥ c[i,j-1]

200

例 (p33)

	j	0	1	2	3	4	5	6
i	y _j	B	D	C	A	B	A	
0	x _i	0	0	0	0	0	0	0
1	A	0	↑0	↑0				
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

i=1, j=2 のとき
x_i≠y_j, c[i-1,j] ≥ c[i,j-1]

201

例 (p33)

	j	0	1	2	3	4	5	6
i	y _j	B	D	C	A	B	A	
0	x _i	0	0	0	0	0	0	0
1	A	0	↑0	↑0	↑0			
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

i=1, j=3 のとき
x_i≠y_j, c[i-1,j] ≥ c[i,j-1]

202

例 (p33)

<i>j</i>	0	1	2	3	4	5	6
<i>i</i>	<i>y_j</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
0	<i>x_i</i>	0	0	0	0	0	0
1	<i>A</i>	0	↑0	↑0	↖1		
2	<i>B</i>	0					
3	<i>C</i>	0					
4	<i>B</i>	0					
5	<i>D</i>	0					
6	<i>A</i>	0					
7	<i>B</i>	0					

i=1, *j*=4 のとき
x_i=*y_j*

203

例 (p33)

<i>j</i>	0	1	2	3	4	5	6
<i>i</i>	<i>y_j</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
0	<i>x_i</i>	0	0	0	0	0	0
1	<i>A</i>	0	↑0	↑0	↑0	↖1	←1
2	<i>B</i>	0					
3	<i>C</i>	0					
4	<i>B</i>	0					
5	<i>D</i>	0					
6	<i>A</i>	0					
7	<i>B</i>	0					

i=1, *j*=5 のとき
x_i≠*y_j*, *c*[*i*-1, *j*] < *c*[*i*, *j*-1]

204

例 (p33)

<i>j</i>	0	1	2	3	4	5	6
<i>i</i>	<i>y_j</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
0	<i>x_i</i>	0	0	0	0	0	0
1	<i>A</i>	0	↑0	↑0	↑0	↖1	←1
2	<i>B</i>	0					
3	<i>C</i>	0					
4	<i>B</i>	0					
5	<i>D</i>	0					
6	<i>A</i>	0					
7	<i>B</i>	0					

i=1, *j*=5 のとき
x_i=*y_j*

205

例 (p33)

<i>j</i>	0	1	2	3	4	5	6
<i>i</i>	<i>y_j</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
0	<i>x_i</i>	0	0	0	0	0	0
1	<i>A</i>	0	↑0	↑0	↑0	↖1	←1
2	<i>B</i>	0	↖1	←1	←1	↑1	↖2
3	<i>C</i>	0					
4	<i>B</i>	0					
5	<i>D</i>	0					
6	<i>A</i>	0					
7	<i>B</i>	0					

206

例 (p33)

<i>j</i>	0	1	2	3	4	5	6
<i>i</i>	<i>y_j</i>	<i>B</i>	<i>D</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>A</i>
0	<i>x_i</i>	0	0	0	0	0	0
1	<i>A</i>	0	↑0	↑0	↑0	↖1	←1
2	<i>B</i>	0	↖1	←1	←1	↑1	↖2
3	<i>C</i>	0	↑1	↑1	↖2	←2	↑2
4	<i>B</i>	0	↖1	↑1	↑2	↑2	↖3
5	<i>D</i>	0	↑1	↖2	↑2	↑2	↑3
6	<i>A</i>	0	↑1	↑2	↑2	↖3	↑3
7	<i>B</i>	0	↖1	↑2	↑2	↑3	↖4

207

アルゴリズムのステップ4:
最適解を構成する

208

```

PRINT-LCS( $b, X, i, j$ )
  if  $i = 0$  or  $j = 0$ 
    then return
    then PRINT-LCS( $b, X, i-1, j-1$ )
    print  $x_i$ 
  elseif  $b[i, j] = "\uparrow"$ 
    then PRINT-LCS( $b, X, i-1, j$ )
  else PRINT-LCS( $b, X, i, j-1$ )
    
```

209

例 (p33)

j	0	1	2	3	4	5	6	
i	y_j	B	D	C	A	B	A	
0	x_i	0	0	0	0	0	0	
1	A	0	$\uparrow 0$	$\uparrow 0$	$\uparrow 0$	$\nwarrow 1$	$\leftarrow 1$	$\nwarrow 1$
2	B	0	$\nwarrow 1$	$\leftarrow 1$	$\leftarrow 1$	$\uparrow 1$	$\nwarrow 2$	$\leftarrow 2$
3	C	0	$\uparrow 1$	$\uparrow 1$	$\nwarrow 2$	$\leftarrow 2$	$\uparrow 2$	$\uparrow 2$
4	B	0	$\nwarrow 1$	$\uparrow 1$	$\uparrow 2$	$\uparrow 2$	$\nwarrow 3$	$\leftarrow 3$
5	D	0	$\uparrow 1$	$\nwarrow 2$	$\uparrow 2$	$\uparrow 3$	$\nwarrow 3$	$\uparrow 3$
6	A	0	$\uparrow 1$	$\uparrow 2$	$\uparrow 2$	$\uparrow 3$	$\uparrow 3$	$\nwarrow 4$
7	B	0	$\nwarrow 1$	$\uparrow 2$	$\uparrow 2$	$\uparrow 3$	$\nwarrow 4$	$\uparrow 4$

210

コードの改善

- $b[i, j]$ を利用しない
 - $c[i, j]$ の矢印の方向は $O(1)$ で求められる
 - 最適解の構成に要する時間: $O(m+n)$
 - メモリ量の節約: $\Theta(mn)$
- $c[i, j]$ のメモリ量を削減する
 - 直前の行 $c[i-1, j]$ のみを覚えておけばよい
 - 最適解の値(長さ)のみを求めればよい場合
 - メモリ量の節約: $\Theta(mn)$

j	0	1	2	3	4	5	6
i	y_j	B	D	C	A	B	A
0	x_i						
1	A						
2	B						
3	C	0	$\uparrow 1$	$\uparrow 1$	$\nwarrow 2$	$\leftarrow 2$	$\uparrow 2$
4	B	0	$\nwarrow 1$	$\uparrow 1$			
5	D						
6	A						
7	B						

211

演習: 以下のLCSを求めよ

j	0	1	2	3	4	5	6
i	y_j	B	C	D	A	C	A
0	x_i	0	0	0	0	0	0
1	A						
2	C						
3	B						
4	D						
5	D						
6	A						
7	C						

212

演習: 以下のLCSを求めよ

j	0	1	2	3	4	5	6	
i	y_j	B	C	D	A	C	A	
0	x_i	0	0	0	0	0	0	
1	A	0	$\uparrow 0$	$\uparrow 0$	$\uparrow 0$	$\nwarrow 1$	$\leftarrow 1$	$\nwarrow 1$
2	C	0	$\uparrow 0$	$\nwarrow 1$	$\leftarrow 1$	$\uparrow 1$	$\nwarrow 2$	$\leftarrow 2$
3	B	0	$\nwarrow 1$	$\uparrow 1$	$\uparrow 1$	$\uparrow 1$	$\uparrow 1$	$\uparrow 2$
4	D	0	$\uparrow 1$	$\uparrow 1$	$\nwarrow 2$	$\leftarrow 2$	$\leftarrow 2$	$\uparrow 2$
5	D	0	$\uparrow 1$	$\uparrow 1$	$\nwarrow 2$	$\uparrow 2$	$\uparrow 2$	$\uparrow 2$
6	A	0	$\uparrow 1$	$\uparrow 1$	$\uparrow 2$	$\nwarrow 3$	$\leftarrow 3$	$\nwarrow 3$
7	C	0	$\uparrow 1$	$\nwarrow 2$	$\uparrow 2$	$\uparrow 3$	$\nwarrow 4$	$\leftarrow 4$

213

例題:
最適二分探索木
(簡単バージョン)

214

問題の定義

入力

- n 個の重み w_i ($1 \leq i \leq n$)が与えられる

出力

- n 個の葉ノードがある二分探索木中で、 $\sum_i w_i \times \text{depth}(i)$ を最小とする二分木(最適二分探索木).
ただし $\text{depth}(i)$ は左から i 番目の葉の深さ

問題の直感的なイメージ

- 重みが探索される頻度に対応、データを二分木に整理するが、探索コストを少なくしたい

$(1 + 2) \times 4 + 3 \times 3 + (4 + 9 + 3) \times 2 = 53$

215

部分構造最適性

- 区間 $[i, j]$ に関して、要素 $i, i + 1, \dots, j$ の最適二分探索木の左部分木が要素 $i, i + 1, \dots, k$ を、右部分木が要素 $k + 1, k + 2, \dots, j$ を含むとする
- この時、左右の部分木は、対応する要素の最適二分探索木になっているはず
- $c_{i,j}$ を、最適解の値とすると、

$$c_{i,j} = c_{i,k} + c_{k+1,j} + \sum_{l=i}^j w_l$$

216

最適解を与える漸化式

- $c_{i,j} = \min_{i \leq k < j} \{c_{i,k} + c_{k+1,j}\} + W_{i,j}$
- ただし、 $W_{i,j} = \sum_{l=i}^j w_l, c_{i,i} = 0$ とする。

217

演習

- 重みを 1, 2, 3, 4, 9, 3 とする
- $c_{i,j}$ の表を埋めよ

						$W_{i,j}$						
			6	1								
		j	4	5	19	21	2	3	i			
		3	10	18	19	4						
		2	6	9	16	16	5					
		1	3	5	7	13	12					

						$c_{i,j}$						
			6	1								
		j	4	5								
		3										
		2										
		1										

$c_{i,j} = \min_{i \leq k < j} \{c_{i,k} + c_{k+1,j}\} + W_{i,j}$
 ただし、 $W_{i,j} = \sum_{l=i}^j w_l, c_{i,i} = 0$ とする。

218

演習 (解答)

- 重みを 1, 2, 3, 4, 9, 3 とする
- $c_{i,j}$ の表を埋めよ

						$W_{i,j}$						
			6	1								
		j	4	5	19	21	2	3	i			
		3	10	18	19	4						
		2	6	9	16	16	5					
		1	3	5	7	13	12					

						$c_{i,j}$						
			6	1								
		j	4	5	53	47	2	3	i			
		3	19	32	38	4						
		2	9	14	23	28	5					
		1	3	5	7	13	12					

219

アルゴリズムの評価

- 計算量
- 3重ループなので $O(n^3)$
- メモリ消費量
- $\Theta(n^2)$

220