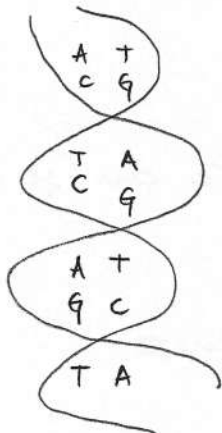


DINAMIČKO PROGRAMIRANJE

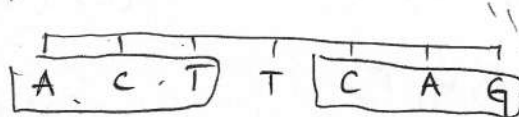
- PROBLEM: uspolediti sličnost DNA dva organizma



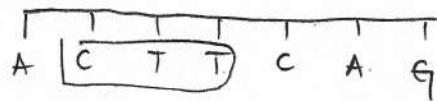
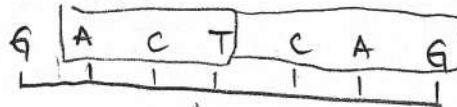
- razmotati polinukleotidne lance
građene od 4 vrste nukleotida

A - adenin }
T - timin }
C - citozin }
G - gvanin }

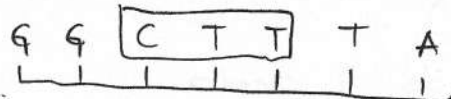
odgovor: koliko je dugačak najdulji zajednički podniz?



6



3



Problem:

Za dana 2 niza

$$X = \langle x_1, \dots, x_m \rangle ; Y = \langle y_1, \dots, y_n \rangle$$

treba pronaći zajednički podniz od X i Y tako da je njegova dužina najveća.

skraćeno: LCS - engl. longest common subsequence

Primjer:

$$X = \langle A, C, T, T, C, A, G \rangle$$

$$lcs(X, Y) = 6$$

$$Y = \langle G, A, C, T, C, A, G, T \rangle$$

$$lcs(X, U) = 3$$

$$U = \langle G, G, C, T, T, T \rangle$$

$$LCS(X, Y) = \langle A, C, T, C, A, G \rangle$$

Naivno rješenje:

- proći svaki podniz od X je li podniz od Y
- uzeti najdulji

Algoritam:

Ulaz: X, Y

Izlaz: $z = LCS(X, Y)$

$c = 0$

Za svaki podniz S od X

if S podniz od Y onda

then if $|S| > c$ then

$z \leftarrow S ; c \leftarrow |S|$

end if

end if

koj za svaki Vрати z

Složenost algoritma???

$$m = |X|, n = |Y|$$

Koliko ima podnizova od X ?

2^m podnizova

Koliko "košta" provjera je li S podskup od Y

$$O(n)$$

Odgovor: $O(n 2^m)$

$$|X|, |Y| \approx 12650000 \text{ (kromosom 5)}$$

$$\text{Broj koraka: } 12650000 \times 2^{12650000} \approx \underline{\underline{10^{10^6}}}$$

$$\text{Broj atoma u svemiru: } \underline{\underline{10^{80}}}$$

Ideja:

- koristiti tehniku dinamičkog programiranja
- rješenje potproblema koristiti za rješenje problema

Oznake:

$$X_i = \langle x_1, \dots, x_i \rangle \text{ prefiks od } X$$

$$Y_i = \langle y_1, \dots, y_i \rangle \quad \text{---} \quad \text{---} \quad Y$$

$$f[i, j] := \text{lcs}(X_i, Y_j)$$

$$f[m, n] := \text{lcs}(X, Y)$$

primjer:

$$X_7 = Y_6$$

A	C	T	T	C	A	G
G	A	C	T	C	A	G

$$Z_6 = X_7 = Y_7$$

A	C	T	T	C	A
G	A	C	T	C	A

$$Z_5 = \text{LCS}(X_6, Y_6)$$

$$X_7 \neq Y_6$$

A	C	T	T	C	A	G
G	G	C	T	T	C	A

$$Z_3 \neq X_7$$

$$Z = \text{LCS}(X_6, Y)$$

$$Z_3 \neq Y_6$$

A	C	T	T	C	A	G
G	G	C	T	G	C	A

$$Z = \text{LCS}(X, Y_5)$$

zaključak:

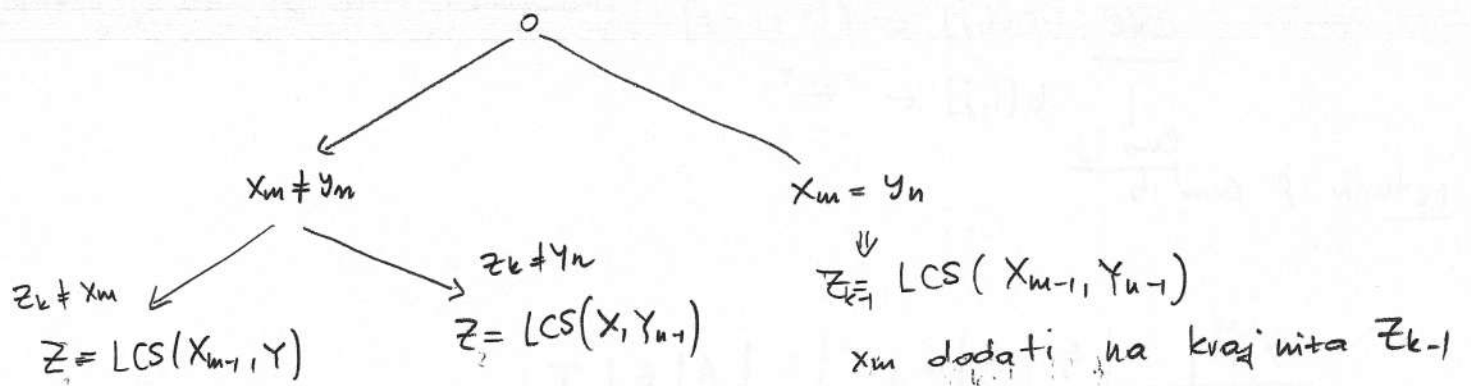
zorem (o optimalnoj podstrukturi LCS problema)

Ako su $X = \langle x_1, \dots, x_m \rangle$ i $Y = \langle y_1, \dots, y_n \rangle$ nizovi i $Z = \langle z_1, \dots, z_k \rangle$ koji LCS od X i Y . Tada vrijede sljedeće tvrdnje:

1. Ako je $x_m = y_n$ onda je $z_k = x_m = y_n$ i $Z_{k-1} = \text{LCS}(X_{m-1}, Y_{n-1})$
2. Neka je $x_m \neq y_n$. Ako je $z_k \neq x_m$ onda je $Z = \text{LCS}(X_{m-1}, Y)$
3. Neka je $x_m \neq y_n$. Ako je $z_k \neq y_n$ onda je $Z = \text{LCS}(X, Y_{n-1})$

izvor: (Leiserson, Rivest, Stein 2nd edition, 351. str)

Interpretacija rekursa:



izračuni će biti
duljini podniza

$$C[i, j] = \begin{cases} 0 & \text{ako je } i=0, j=0 \\ C[i-1, j-1] + 1 & i, j > 0 \text{ } x_i = y_j \\ \max(C[i, j-1], C[i-1, j]) & i, j > 0 \text{ } x_i \neq y_j \end{cases}$$

- najprije ćemo izračunati duljinu najdužeg zajedničkog podniza

LCS-LENGTH(X, Y)

$m \leftarrow X.length$

$n \leftarrow Y.length$

for $i \leftarrow 1$ to m

do $C[i, 0] \leftarrow 0$

end for

for $j \leftarrow 1$ to n

do $C[0, j] \leftarrow 0$

end for

for $i \leftarrow 1$ to m

do for $j \leftarrow 1$ to n

do if $x_i = y_j$

then $C[i, j] \leftarrow C[i-1, j-1] + 1$

$b[i, j] \leftarrow \nwarrow$

else if $C[i-1, j] \geq C[i, j-1]$

then $C[i, j] \leftarrow C[i-1, j]$

```

    if b[i,j] ← "↑"
    else f[i,j] ← f[i,j-1]
    if b[i,j] ← "←"
    end if
return f and b

```

X \ Y		G	A	C	T	C	A	G	T
	0	0	0	0	0	0	0	0	0
A	0	0	1	1	1	1	1	1	1
C	0	0	1	2	2	2	2	2	2
T	0	0	1	2	3	3	3	3	3
T	0	0	1	2	3	3	3	3	4
C	0	0	0	2	3	4	4	4	4
A	0	0	1	2	3	4	5	5	5
G	0	1	1	2	3	4	5	6	6

$$\text{LCS}(X, Y) = f[7,8] = 6$$

rekonstrukcija najduljeg zajedničkog podniza

PRINT LCS(b, X, i, j)

if i=0 or j=0
then return

if b[i,j] = "↖"

then PRINT-LCS(b, X, i-1, j-1)
print: X_i

else if b[i,j] = "↑"

then PRINT-LCS(b, X, i-1, j)

else PRINT-LCS(b, X, i, j-1)

rezultat na prethodnom primjeru: ACTCAG

rijeme izračunavanja: LCS-LENGTH - $\Theta(mn)$

PRINT-LCS - $\mathcal{O}(m+n)$

- LCS - primjer optimizacijskog problema

- optimizacijski problem - svako rješenje ima pridruženu vrijednost
cijl je pronaći rješenje s najvećom / najmanjom
vrijednošću

- dinamičko programiranje koristimo u rješavanju uglavnom
optimizacijskih problema

- ključni koraci:

1. Karakterizirati strukturu optimalnog rješenja
2. Rekursivno definirati vrijednost optimalnog rješenja
3. Izračunati vrijednost optimalnog rješenja (koristeći
bottom-up pristup)
4. Rekonstruirati optimalno rješenje na osnovu izračunatih
informacija

Primjer (LCS)

1. Teorem o optimalnoj podstrukturi LCS problema
2. Rekursivna relacija za računanje optimalne vrijednosti
3. LCS-LENGTH algoritam i tablica
4. PRINT-LCS algoritam (i tablica)

Ulančano množenje matrica

Problem:

ulaz: niz matrica A_1, \dots, A_n , $n \in \mathbb{N}$ za koje je definiran produkt

$$A_1 A_2 \dots A_n$$

izlaz: izračunati produkt $A_1 A_2 \dots A_n$ koristeći najmanji mogući broj operacija množenja skalara

- asociativnost množenja matrica:

$$(A_1 (A_2 (A_3 A_4)))$$

$$(A_1 ((A_2 A_3) A_4))$$

$$((A_1 A_2) (A_3 A_4))$$

$$((A_1 (A_2 A_3)) A_4)$$

$$(((A_1 A_2) A_3) A_4)$$

- uočimo da raspored zagradama ima utjecaj na broj operacija

Primjer:

$$\begin{array}{ccccc} A_1 & & A_2 & & A_3 \\ 10 \times 100 & \times & 100 \times 5 & \times & 5 \times 50 \end{array}$$

dva načina: $((A_1 A_2) A_3)$

$$(A_1 \times A_2)$$

$$10 \cdot 100 \cdot 5 = \underline{5000}$$

$$(A_1 \times A_2) \times A_3$$
$$10 \times 5 \quad 5 \times 50$$

$$10 \cdot 5 \cdot 50 = \underline{2500}$$

Ukupno: 7500 op.

$$A_1 (A_2 A_3)$$

$$(A_2 \times A_3)$$

$$100 \times 5 \cdot 50 = \underline{25000}$$

$$A_1 \times (A_2 \times A_3)$$

$$100 \times 100 \quad 100 \times 50$$

$$10 \cdot 100 \cdot 50 = \underline{50000}$$

Ukupno: 75000 op.

Izljuček: prvi pristup je 10 puta brži.

Znake:

$$\begin{matrix} A_1 & A_2 & & A_n \\ p_0 \times p_1 & p_1 \times p_2 & \dots & p_{n-1} \times p_n \end{matrix}$$

matrica A_i ima
dimenziji

roj različnih načina postavljanja zagrada

$$(A_1 \dots A_k) (A_{k+1} \dots A_n)$$

$$k = 1, \dots, n-1$$

broj načina
ovdje



broj načina
ovdje

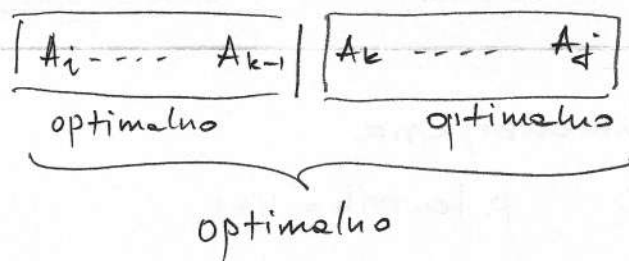
$$P(n) = \begin{cases} 1 & , n=1 \\ \sum_{k=1}^{n-1} P(k)P(n-k) & , n \geq 2 \end{cases}$$

$$(A_1)$$

Može se pokazati da je $P(n) \in \Omega(2^n)$ (D.Z).

Brute-force metoda ne prolazi!

Struktura optimalnog postavljanja zagrada



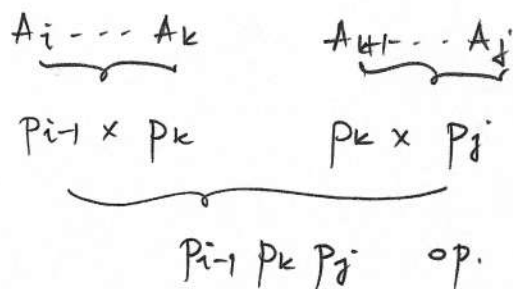
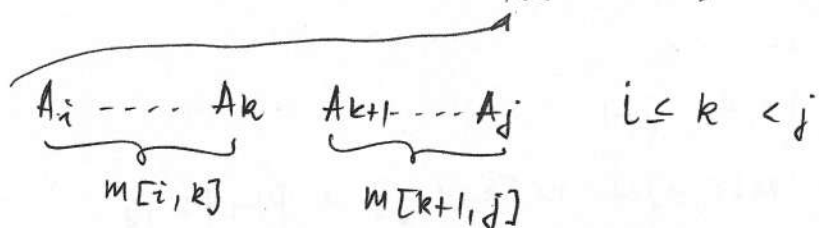
Rekurzija:

$m[i, j] \leftarrow$ minimalan broj skal. množenja za računanje produkta $A_i \dots A_j$ $1 \leq i \leq j \leq n$

rješenje (ovo što tražimo) je $m[1, n]$

$i = j \Rightarrow A_i \dots A_i \Rightarrow m[i, i] = 0 \quad 1 \leq i \leq n$

$i < j \Rightarrow A_i \dots A_j \Rightarrow m[i, j] = m[i, k] + m[k+1, j] + p_{i-1} p_k p_j$



Rekurzivna relacija:

$$m[i, j] = \begin{cases} 0 & i = j \\ \min_{i \leq k < j} \{ m[i, k] + m[k+1, j] + p_{i-1} p_k p_j \} & i < j \end{cases}$$

oznaka: neka je $s[i, j] = \bar{k}$ takav da je

$$m[i, j] = m[i, \bar{k}] + m[\bar{k}+1, j] + p_{i-1} p_{\bar{k}} p_j$$

pristup računanja $m[i,j]$ je tabularizacija (bottom-up)
 pristup kao kod LCS-a.

• broj moženja različitih n dimenzijama

$$p = \langle p_0, p_1, \dots, p_n \rangle \quad p.\text{length} = n+1$$

MATRIX-CHAIN-ORDER (p)

$$n \leftarrow p.\text{length} - 1$$

for $i \leftarrow 1$ to n

$$m[i,i] \leftarrow 0$$

for $l \leftarrow 2$ to n

for $i \leftarrow 1$ to $n-l+1$

$$j \leftarrow i+l-1$$

$$m[i,j] \leftarrow \infty$$

for $k \leftarrow i$ to $j-1$

$$q \leftarrow m[i,k] + m[k+1,j] + p_{i-1} p_k p_j$$

if $q < m[i,j]$

$$m[i,j] \leftarrow q$$

$$s[i,j] \leftarrow k$$

end if

end for

end for

end for

return m, s

$$A_i \dots A_j$$

$$p_{i-1} \dots p_j$$

$$\text{ima ih } j - (i-1) = \underline{j-i+1}$$

dužina
↓

$$A_i \dots \underbrace{A_{i+l-1}}_j$$

Primer:

matrica	A_1	A_2	A_3	A_4	A_5	A_6
dim	30×35	35×15	15×5	5×10	10×20	20×25

i \ j	1	2	3	4	5	6
1	0	15750	7875	9375	11875	15125
2		0	2625	4375	7125	10500
3			0	750	2500	5375
4				0	1000	3500
5					0	5000
6						0

i \ j	1	2	3	4	5	6
1	x	x	1	3	3	3
2		x	2	3	3	3
3			x	3	3	3
4				x	4	5
5					x	4
6						x

$$m[1,2] = m[1,1] + m[2,2] + p_0 p_1 p_2 = 30 \cdot 35 \cdot 15 = 15750$$

$$m[2,3] = m[2,2] + m[3,3] + p_1 p_2 p_3 = 35 \cdot 15 \cdot 5 = 2625$$

$$m[3,4] = \dots + p_2$$

$$m[4,5] = \dots$$

$$m[5,6] = m[5,5] + m[6,6] + p_4 p_5 p_6 = 10 \cdot 20 \cdot 25 = 5000$$

$$m[1,3] = m[1,1] + m[2,3] + p_0 p_1 p_3 = 0 + 2625 + 5250 = 7875$$

$$m[1,3] = \min \begin{cases} m[1,1] + m[2,3] + p_0 p_1 p_3 = 0 + 2625 + 5250 = 7875 \\ m[1,2] + m[3,3] + p_0 p_2 p_3 = 15750 + 0 + 2250 = 18000 \end{cases}$$

$$s[1,3] = 1$$

Konstrukcija optimalnog rješenja:

PRINT-OPTIMAL-PARENS (s, i, j)

if $i == j$

print A_i

else print "("

PRINT-OPTIMAL-PARENS ($s, i, s[i,j]$)

PRINT-OPTIMAL-PARENS ($s, s[i,j]+1, j$)

print ")"

end if

Poziv:

PRINT-OPTIMAL-PARENS (S, 1, 6)

$((1) (2\ 3) (4\ 5\ 6))$

$(A_1 (A_2 A_3)) ((A_4 A_5) A_6)$

/rijeme izvršavanja:

MATRIX-CHAIN-ORDER:

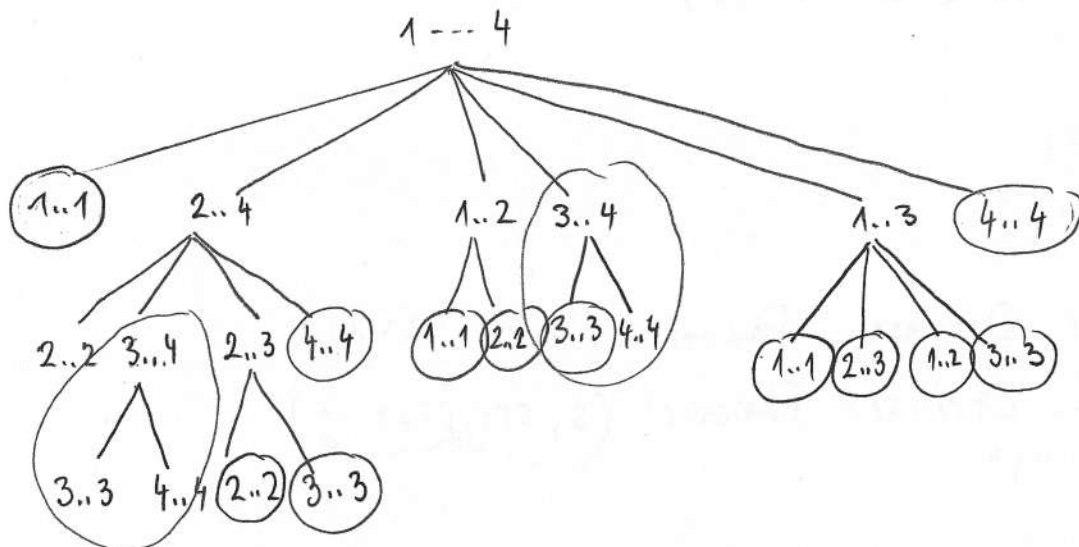
3 for petlje

$$T(n) = \sum_{l=2}^n \sum_{i=1}^{n-l+1} \sum_{k=i}^{i+l-2} \Theta(1) = O(n^3)$$

D.t. $\Theta(n^3)$ (odnosno $T(n) \in \mathcal{O}(n^3)$)

prostorna složenost: $\Theta(n^2)$

Razmotrimo rekursivan pristup (nije tabularizacija)



RECURSIVE-MATRIX-CHAIN (p_i, j)

```

if  $i=j$ 
    return 0
 $m[i, j] \leftarrow \infty$ 
for  $k \leftarrow i$  to  $j-1$ 
     $q = \text{RECURSIVE-MATRIX-CHAIN}(p_i, k)$ 
        +  $\text{RECURSIVE-MATRIX-CHAIN}(p, k+1, j)$ 
        +  $p_i - p_k + p_j$ 
    if  $q < m[i, j]$ 
         $m[i, j] \leftarrow q$ 
    end if
end for
return  $m[i, j]$ 

```

RECURSIVE-MATRIX-CHAIN (RMC)

Vrijeme izvršavanja: (dobra metoda)

$$T(1) \geq 1$$

$$T(n) \geq 1 + \sum_{k=1}^{n-1} (T(k) + T(n-k) + 1)$$

ekvivalentno

$$T(n) \geq 2 \sum_{i=1}^{n-1} T(i) + n$$

(stabilna rekursija + metoda supstitucije)

Pretp: $T(n) = \Omega(2^n)$

$$T(n) \geq 2^{n-1}, \text{ uel}$$

Baza: $T(1) \geq 2^{1-1} = 2^0 = 1$

Pretpostavka: $T(k) \geq 2^{k-1}, k \leq n, 1$

Korak: $T(n+1) \geq 2 \sum_{i=1}^{(n+1)-1} T(i) + n+1$

$$\begin{aligned}
&\geq 2 \sum_{i=1}^n 2^{i-1} + n+1 \\
&= 2 \sum_{i=0}^{n-1} 2^i + (n+1) \\
&= 2(2^n - 1) + (n+1) \\
&= 2^{n+1} - 2 + n+1 = 2^{n+1} + (n-1) \\
&\geq \underline{\underline{2^n}} [= 2^{(n+1)-1}] \Rightarrow T(n) = \underline{\underline{\Omega(2^n)}}
\end{aligned}$$

-ukoliko "pamtimo" rješenja potproblema koji se preklapaju dolazimo do pristupa koji se zove memoizacija

MEMOIZED-MATRIX-CHAIN (p)

$n \leftarrow p.length - 1$

```

for i ← 1 to n
  for j ← 1 to n
    m[i,j] ← ∞
  end for
end for

```

return LOOKUP-CHAIN (m, p, 1, n)

LOOKUP-CHAIN (m, p, i, j)

```

if m[i,j] < ∞
  return m[i,j]

```

end if

```

if i == j
  m[i,j] ← 0

```

```

else
  for k ← i to j-1

```


$$\begin{aligned}
 & q \leftarrow \text{lookup-CHAIN}(m, p, i, k) \\
 & \quad + \text{lookup-CHAIN}(m, p, k+1, j) \\
 & \quad + p_i + p_k + p_j
 \end{aligned}$$

$$\begin{aligned}
 & \text{if } q < m[i, j] \\
 & \quad | \quad m[i, j] \leftarrow q
 \end{aligned}$$

end if

return $m[i, j]$

Vrijeme izvršavanja: $O(n^3)$??? (analiza D.2)