

۳۳- می‌خواهیم با وارد کردن مقادیر ۱ و ۲ و ۳ به هر ترتیب دلخواه در یک درخت تهی دودویی جستجو (Null Binary Search Tree) یک درخت دودویی جستجو با ۳ گره بسازیم. چند درخت دودویی جستجوی متفاوت ممکن است ساخته شود؟ (مهندسی کامپیوتر - دولتی ۷۶)

(۱) ۴ درخت (۲) ۳ درخت (۳) ۶ درخت (۴) ۵ درخت

۳۸- اگر  $T$  یک درخت جستجوی باینری به صورت زیر باشد که در هر گره آن یک عدد صحیح ذخیره شده است چهارمین کوچکترین عنصر آن در کدام گره قرار دارد؟ (علوم کامپیوتر - دولتی ۸۶)

۴۷- می‌خواهیم یک درخت دودویی جستجو با عنصر  $a_1 < a_2 < \dots < a_6$  بسازیم تا متوسط عمق عناصر در

آن کمینه شود. اگر  $p_i$  احتمال  $a_i$  باشد، متوسط عمق برابر  $\sum_{i=1}^6 p_i \text{depth}(q_i)$  تعریف می‌شود. اگر  $p_1 = \frac{2}{7}$

و  $p_i = \frac{1}{7}$  برای  $2 \leq i \leq 6$  باشد، متوسط عمق درخت بهینه چند است؟ (عمق ریشه صفر فرض می‌شود)

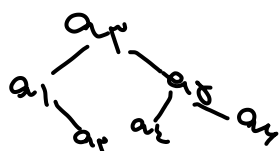
(علوم کامپیوتر - دولتی ۸۲)

$\frac{11}{7}$  (۴)

$\frac{10}{7}$  (۳)

$\frac{9}{7}$  (۲)

$\frac{8}{7}$  (۱)



## طولانی ترین زیر دنباله مشترک (LCS) دو دنباله:

مساله LCS کاربردهای متعددی دارد، از جمله مقایسه متن، تراز کردن توالی DNA، و تشخیص سرقت ادبی. معمولاً برای تعیین شباهت یا عدم تشابه بین دنباله ها و شناسایی الگوها یا ساختارهای مشترک استفاده می شود.

مثلا در برخی کاربردهای بیولوژیکی لازم است DNA چند ارگانیزم با هم مقایسه شود. یک DNA از زنجیره ای از مولکول های پایه ای تشکیل شده است. برخی از آنها عبارتند از آدنین (A)، گوانین (G)، سیتوسین (S) و تیمین (T). بنابراین یک رشته را می توان کلمه ای روی  $\{A, G, S, T\}$  در نظر گرفت. مثلاً

$S_1 = \text{ACCGGTCGAGTGC} \underline{\text{GCGGAAGCCGGCCGAA}}$

$S_2 = \text{GTCGTTCGGAATGCCGTTGCTCTGTAA}$

یکی از اهداف مقایسه دو رشته تعیین میزان همانندی آنهاست. لذا دنبال رشته سومی هستیم که مولکول پایه آن در هر دو رشته ظاهر شده باشد ولی لزومی ندارد متوالی باشند. در مثال فوق رشته مشترک بصورت زیر است:

$\underline{\text{GTCGTTCGGAAGCCGGCCGAA}}$

- How similar are these two species?



NA:

AGCCCTAAGGGCTACCTAGCTT



DNA:

GACAGCCTACAAGCGTTAGCTTG

- Pretty similar, their DNA has a long common subsequence:

AGCCTAAGCTTAGCTT

پیشوند  $i$ ام  $X$ :  $i$  حرف پشت سر هم از ابتدای  $X$  را پیشوند  $i$ ام  $X$  گویند و با  $X_i$  نشان می دهند.

زیر دنباله: گوئیم  $Z$  زیر دنباله ای از  $X$  است اگر

$$\exists i_1, i_2, \dots, i_k \mid i_1 < i_2 < \dots < i_k : \forall j : 1 \leq j \leq k \ z_j = x_{i_j}$$

مثال:

$X = \langle A, B, B, A, D, A, B, F \rangle$

$Z = \langle B, A, A, F \rangle$

$$\begin{cases} i_1 = 2 \Rightarrow z_1 = x_2 \\ i_2 = 4 \Rightarrow z_2 = x_4 \\ i_3 = 6 \Rightarrow z_3 = x_6 \\ i_4 = 8 \Rightarrow z_4 = x_8 \end{cases}$$

نکته: هر رشته  $m$  حرفی دارای  $2^m$  زیر دنباله است.

در این مسئله در ورودی دو رشته به طول های  $n, m$  وجود دارد. که هدف مسئله بدست آوردن طولانی ترین زیر دنباله مشترک می باشد.

ورودی:

دو دنباله  $X$  و  $Y$  که اولی به طول  $m$  و دومی به طول  $n$  می باشد.

خروجی: طول طولانی ترین زیر دنباله مشترک  $X$  و  $Y$

## Brute-force LCS algorithm

Check every subsequence of  $x[1 \dots m]$  to see if it is also a subsequence of  $y[1 \dots n]$ .

### Analysis

- Checking =  $O(n)$  time per subsequence.
- $2^m$  subsequences of  $x$  (each bit-vector of length  $m$  determines a distinct subsequence of  $x$ ).

Worst-case running time =  $O(n2^m)$   
= exponential time.

رویکرد برنامه ریزی پویا:

۱- طول طولانی ترین زیر دنباله مشترک  $X_i$  و  $Y_j$   $c[i,j] =$

۲-  $c[i,0]=c[0,j]=0$  ;  $\forall i,j$  (بدیهی) وقتی یکی از رشته ها تهی باشد هیچ اشتراکی وجود نخواهد

داشت)

۳-  $c[m,n]$  = جواب

۴- رابطه بازگشتی:

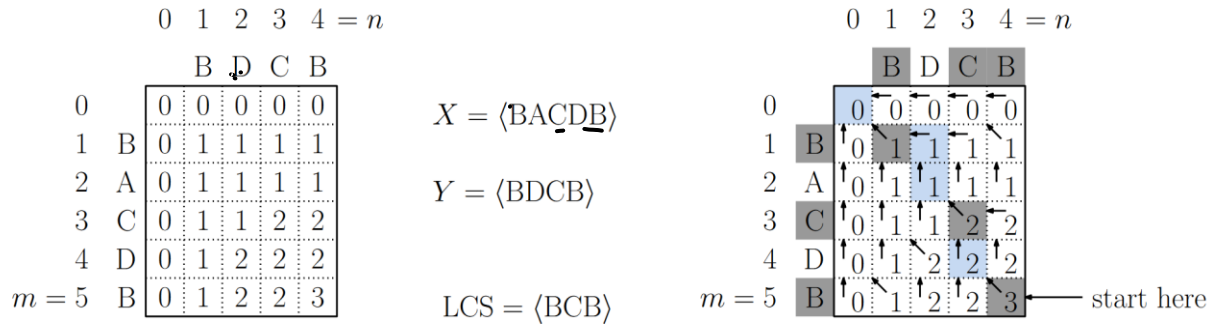
$$c[i,j] = \begin{cases} c[i-1,j-1]+1 & ; x_i = y_j \\ \max\{c[i-1,j], c[i,j-1]\} & ; else \end{cases}$$

*LCS – Length*( $X, Y$ )

```
1   $m \leftarrow \text{length}[X]$ 
2   $n \leftarrow \text{length}[Y]$ 
3  for  $i \leftarrow 1$  to  $m$ 
4      do  $c[i, 0] \leftarrow 0$ 
5  for  $j \leftarrow 0$  to  $n$ 
6      do  $c[0, j] \leftarrow 0$ 
7  for  $i \leftarrow 1$  to  $m$ 
8      do for  $j \leftarrow 1$  to  $n$ 
9          do if  $x_i = y_j$ 
10             then  $c[i, j] \leftarrow c[i - 1, j - 1] + 1$ 
11                  $b[i, j] \leftarrow \text{“}\nwarrow\text{”}$ 
12             else if  $c[i - 1, j] \geq c[i, j - 1]$ 
13                 then  $c[i, j] \leftarrow c[i - 1, j]$ 
14                      $b[i, j] \leftarrow \text{“}\uparrow\text{”}$ 
15                 else  $c[i, j] \leftarrow c[i, j - 1]$ 
16                      $b[i, j] \leftarrow \text{“}\leftarrow\text{”}$ 
17  return  $c$  and  $b$ 
```

پیچیدگی زمانی این الگوریتم  $\theta(mn)$  است.

مثال:



مثال:

ورودی: دو دنباله  $X$  به طول  $m$  و  $Y$  به طول  $n$

خروجی: طول طولانی‌ترین زیر دنباله مشترک  $X$  و  $Y$

$X = \langle A, B, B, A, D, A, B, F \rangle$

$Y = \langle B, F, A, F, D, H, B \rangle$

```

procedure print_LCS(X,b,i,j)
  if (i=0) or (j=0) then return endif
  if b[i,j] = '\n' then
    print_LCS(X,b, i-1, j-1)
    write (xi)
  elsif b[i,j] = '→' then
    print_LCS(X,b,i,j-1)
  else
    print_LCS(X,b, i-1,j)
  endif
end.

```

الگوریتم `print_LCS` مسیر مشخص شده به رنگ خاکستری را از آخر به اول طی کرده و دوباره باز می‌گردد (به علت بازگشتی بودن رویه) و در حین بازگشت خانه‌هایی که در آن فلش مورب می‌بیند را چاپ می‌کند. از این رو طولانی‌ترین زیر دنباله مشترک  $X$  و  $Y$  بصورت زیر چاپ خواهد شد:

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