Un-weighted Shortest Path Problem

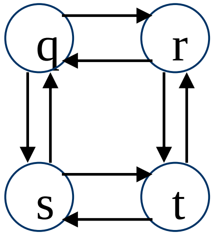
- It exhibits the optimal substructure.
  - Suppose that $u \neq v$ and an optimal path p from u to v must contain an intermediate vertex w.
  - Thus, we can decompose the path p from u to v into p_1 from u to w and p_2 from w to v.
  - If p is optimal path, p_1 and p_2 are also optimal path.
  - Proof:
    - Assume that $p_1 \rightarrow p_2$ is not an optimal path and let $p_1' \rightarrow p_2'$ is an optimal path such that #ofedges($p_1 \rightarrow p_2$) > #ofedges($p_1' \rightarrow p_2'$).
    - Let us replace p_1' by p_1 and replace p_2' by p_2.
    - Then, we have #ofedges($p_1' \rightarrow p_2'$) >= #ofedges($p_1 \rightarrow p_2$).
    - Since #ofedges($p_1 \rightarrow p_2$) > #ofedges($p_1' \rightarrow p_2'$) by our assumption, we have #ofedges($p_1 \rightarrow p_2$) > #ofedges($p_1' \rightarrow p_2'$) >= #ofedges($p_1 \rightarrow p_2$).
    - It contradicts and thus $p_1 \rightarrow p_2$ is an optimal path.

Un-weighted Longest Path Problem

- It does not exhibit optimal substructure.
  - Let’s decompose a longest simple path p from u to v into p_1 from u to w and p_2 from w to v.
  - p_1 may not a longest path from u to w.
  - p_2 may not a longest path from w to v.

Un-weighted Longest Path Problem

- Consider the path q→r→t which is a longest simple path from q to t.
- It is decomposed into q→r and r→t.
- But q→r is not a longest path from q to r. Actually, q→s→t→r is the longest path.
- Thus, the problem lacks optimal structure.
- We cannot assemble a "legal" solution to the problem from solutions to subproblems.
- The subproblems in finding the longest simple path are not independent.



Un-weighted Longest Path Problem

- What do we mean by subproblems being independent?
- The solution to one subproblem does not affect the solution to another subproblem of the same problem.
  - In the previous example, we have the problem of finding a longest simple path from q to t with two subproblems: finding longest simple paths from q to r and from r to t .
  - For the first of these subproblems, we can choose the path q→s→t→r.
  - We can no longer use these vertices in the second subproblem, since the combination of the two solutions to subproblems would yield a path that is not simple.

Longest Common Subsequence

Definitions

- Subsequence

  X=<A,B,C,B,D,A,B>

  → <B,C,D,B>, <A,B,A,B>
- Common subsequence

  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>

  → <B,C,A>

  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Definitions

- Subsequence
  X=<A,$\boxed{B}$,$\boxed{C}$,B,$\boxed{D}$,A,$\boxed{B}$>
  → <$\boxed{B}$,$\boxed{C}$,$\boxed{D}$,$\boxed{B}$>, <A,B,A,B>
- Common subsequence
  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>
  → <B,C,A>
  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Definitions

- Subsequence
  X=<A,B,C,B,D,A,B>
  → <B,C,D,B>, <A,B,A,B>
- Common subsequence
  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>
  → <B,C,A>
  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Definitions

- Subsequence

  X=<A,B,C,B,D,A,B>

  → <B,C,D,B>, <A,B,A,B>
- Common subsequence

  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>

  → <B,C,A>

  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Definitions

- Subsequence

  X=<A,B,C,B,D,A,B>

  → <B,C,D,B>, <A,B,A,B>

- Common subsequence

  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>

  → <B,C,A>

  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Definitions

- Subsequence

  X=<A,B,C,B,D,A,B>

  → <B,C,D,B>, <A,B,A,B>
- Common subsequence

  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>

  → <B,C,A>

  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Definitions

- Subsequence

  X=<A,B,C,B,D,A,B>

  → <B,C,D,B>, <A,B,A,B>
- Common subsequence

  X = <A,B,C,B,D,A,B>, Y = <B,D,C,A,B,A>

  → <B,C,A>

  → <B,C,B,A>, <B,D,A,B> :

Longest common subsequence(LCS)

Longest Common Subsequence(LCS) Problem

- Find a maximun-length common subsequence of $X=<x_1,x_2,\ldots, x_m>$ and $Y=<y_1,y_2,\ldots,y_n>$
- The LCS problem can be solved using dynamic programming.

Characterizing a LCS

- A brute-force approach enumerates all subsequences of X and check each subsequence to see whether it is also a subsequence of Y, keeping track of the longest subsequence we find.
- Each subsequence of X corresponds to a subset of the indices {1,2,…m} of X.
- Because X has 2^m subsequences, it requires exponential time, making it impractical for long sequences.

Optimal Substructure of an LCS

- The natural classes of subproblems correspond to pairs of “prefixes” of the two input sequences.
- To be precise, given a sequence $X=<x_1,x_2,\ldots,x_m>$, we define the i-th prefix of X, for i=1,…,m, as $X_i=<x_1,x_2,\ldots,x_i>$.
- e.g.) If X=<A, B, C, B, D, A, B>, then X_4=<A,B,C,B> and X_0 is the empty sequence.

Optimal Substructure of an LCS

Theorem 15.1

- Let $X=\langle x_1,x_2,\ldots,x_m\rangle$ and $Y=\langle y_1, y_2,\ldots,y_n\rangle$ be sequences, and let $Z=\langle z_1,z_2,\ldots,z_k\rangle$ be any LCS of X and Y

  (1) If $x_m=y_n$, we have $z_k=x_m=y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}
  (2) If $x_m\neq y_n$ and $z_k\neq x_m$ → Z is an LCS of X_{m-1} and Y_n
  (3) If $x_m\neq y_n$ and $z_k\neq y_n$ → Z is an LCS of X_m and Y_{n-1}

- e.g.)

  (1) X=<A,C,D,B>, Y=<A,E,C,F,B>, LCS Z=<A,C,B>

  Z_2=<A,C> is an LCS of <A,C,D>, <A,E,C,F>

  (2) X=<A,C,D,B,E> Y=<A,E,C,B>, LCS Z=<A,C,B>

  Z=<A,C,B> is an LCS of <A,C,D,B>, <A,E,C,B>

Optimal Substructure of an LCS

Theorem 15.1

- Let $X=<x_1,x_2,\ldots,x_m>$ and $Y=<y_1, y_2,\ldots,y_n>$ be sequences, and let $Z=<z_1,z_2,\ldots,z_k>$ be any LCS of X and Y

(1) If $x_m=y_n$, we have $z_k=x_m=y_n$ and Z_{k-1} is an LCS of X_{m-1} and Y_{n-1}

Proof

- If $z_k \neq x_m$, then we could append $x_m=y_n$ to Z to obtain a common subsequence of X and Y of length k+1, contradicting the supposition that Z is a longest common subsequence of X and Y . Thus, $z_k=x_m=y_n$.
- Now, the prefix Z_{k-1} is a length-(k -1) common subsequence of X_{m-1} and Y_{n-1}. We wish to show that it is an LCS.
- Suppose for the purpose of contradiction that there exists a common subsequence W of X_{m-1} and Y_{n-1} with length greater than k-1.
- Then, appending $x_m=y_n$ to W produces a common subsequence of X and Y whose length is greater than k, which is a contradiction.

Optimal Substructure of an LCS

Theorem 15.1

- Let $X=<x_1,x_2,\ldots,x_m>$ and $Y=<y_1, y_2,\ldots,y_n>$ be sequences, and let $Z=<z_1,z_2,\ldots,z_k>$ be any LCS of X and Y

(2) If $x_m \neq y_n$ and $z_k \neq x_m$ → Z is an LCS of X_{m-1} and Y_n

(3) If $x_m \neq y_n$ and $z_k \neq y_n$ → Z is an LCS of X_m and Y_{n-1}

Proof

- If $z_k \neq x_m$, then Z is a common subsequence of X_{m-1} and Y .
- If there were a common subsequence W of X_{m-1} and Y with length greater than k, then W would also be a common subsequence of X_m and Y, contradicting the assumption that Z is an LCS of X and Y.
- The proof is symmetric to (2).

Optimal Substructure of an LCS

- The way that Theorem 15.1 characterizes the longest common subsequences tells us that an LCS of two sequences contains within it an LCS of prefixes of the two sequences.
- Thus, the LCS problem has an optimal-substructure property.

A Recursive Solution

- Theorem 15.1 implies that we should examine either one or two subproblems when finding an LCS of $X=<x_1,x_2,\ldots,x_m>$ and $Y=<y_1, y_2,\ldots,y_n>$.
- If $x_m= y_n$, we must find an LCS of X_{m-1} and Y_{n-1}.
  - Appending $x_m=y_n$ to this LCS yields an LCS of X and Y.
- If $x_m \neq y_n$, then we must solve two subproblems
  - Finding an LCS of X_{m-1} and Y and finding an LCS of X and Y_{n-1}.
  - Whichever of these two is longer is an LCS of X and Y.
- Because these cases exhaust all possibilities, we know that one of the optimal subproblem solutions must appear within an LCS of X and Y .

Overlapping Subproblems

- We can see the overlapping-subproblems property in the LCS problem.
- To find an LCS of X and Y, we may need to find the LCSs of X and Y_{n-1} and of X_{m-1} and Y.
- But each of these subproblems has the subsubproblem of finding an LCS of X_{m-1} and Y_{n-1}.
- Many other subproblems share subsubproblems.

A Recursive Solution

- Let c[i, j] be the length of an LCS of the sequences X_i and Y_j
- If either i = 0 or j = 0, one of the sequences has length 0, and so the LCS has length 0.

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

A Recursive Solution

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

- In this recursive formulation, a condition in the problem restricts which subproblems we may consider.
- When x_i=y_j , we can and should consider the subproblem of finding an LCS of X_{i-1} and Y_{j-1}.
- Otherwise, we instead consider the two subproblems of finding an LCS of X_i and Y_{j-1} and of X_{i-1} and Y_j.
- In the previous dynamic-programming algorithms we have examined—for rod cutting and matrix-chain multiplication—we ruled out no subproblems due to conditions in the problem.

Computing the Length of LCS

- Procedure LCS-LENGTH takes two sequences $X=<x_1,x_2,\ldots,x_m>$ and $Y=<y_1, y_2,\ldots,y_n>$ as inputs.
- It stores the c[i, j] values in a table c[0..m, 0..n], and it computes the entries in row-major order. (That is, the procedure fills in the first row of c from left to right, then the second row, and so on.)
- The procedure also maintains the table b[1..m, 1..n] to help us construct an optimal solution.
- Intuitively, b[i, j] points to the table entry corresponding to the optimal subproblem solution chosen when computing c[i, j].
- The procedure returns the b and c tables where c[m, n] contains the length of an LCS of X and Y .

Computing the Length of LCS

```
LCS-LENGTH(X,Y)
m = X.length
n = Y.length
let b[1..m,1..n] and c[0..m,0..n] be new tables
for i =1 to m
    c[i,0] = 0
for i = 0 to n
    c[0, j] = 0
for i = 1 to m
    for j = 1 to n
        if $x_i == y_j$
            c[i, j] = c[i-1, j-1] + 1
            b[i, j] = "↖ "
        else if c[i-1, j] ≥ c[i, j -1]
            c[i, j] = c[i-1, j]
            b[i, j] = "↑"
        else
            c[i, j] = c[i, j-1]
            b[i, j] = "←"
return c and b
```

Running time is $\theta(mn)$

Computing the Length of LCS

- Let m be the length of sequence X and n be the length of sequence Y.
- Space Complexity is θ(mn).
- Time Complexity is θ(mn).
    - LCS problem has only θ(mn) distinct subproblems.
    - Each table entry takes $\theta(1)$ time to compute.

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

Computing the Length of LCS

	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						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

Computing the Length of LCS

	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						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[1,1] = \max(C[1,0], C[0,1]) = 0 \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{B} = \mathrm{y_1}$

$B[1,1] = \text{"}\uparrow\text{"} \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{B} = \mathrm{y_1} \text{ and } C[0,1] \geq C[1,0]$

Computing the Length of LCS

	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						

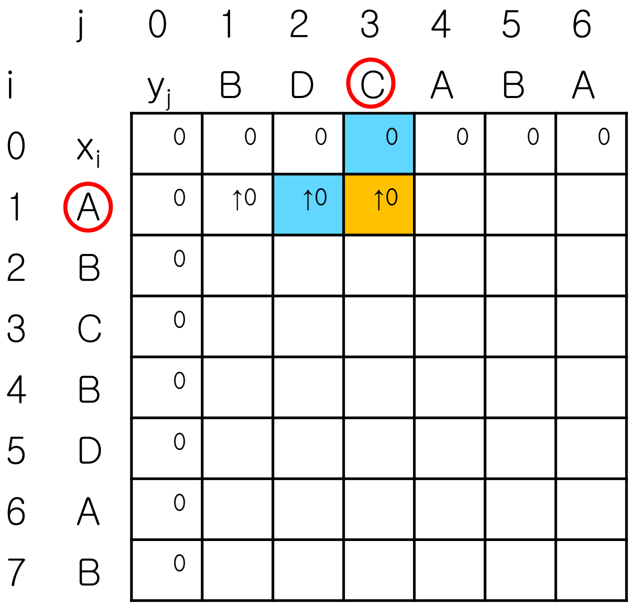
$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } \mathrm{x_i} = \mathrm{y_j} \\ \max(C[i,j-1], C[i-1,j]) & \text{if } i,j > 0 \text{ and } \mathrm{x_i} \neq \mathrm{y_j} \end{cases}$$

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[1,2] = \max(C[1,1], C[0,2]) = 0 \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{D} = \mathrm{y_2}$

$B[1,2] = \text{"}\uparrow\text{"} \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{D} = \mathrm{y_2} \text{ and } C[0,2] \geq C[1,1]$

Computing the Length of LCS



	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						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[1,3] = \max(C[1,2], C[0,3]) = 0 \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{C} = \mathrm{y_3}$

$B[1,3] = \text{"}\uparrow\text{"} \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{C} = \mathrm{y_3} \text{ and } C[0,3] \geq C[1,2]$

Computing the Length of LCS

	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	↖1		
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

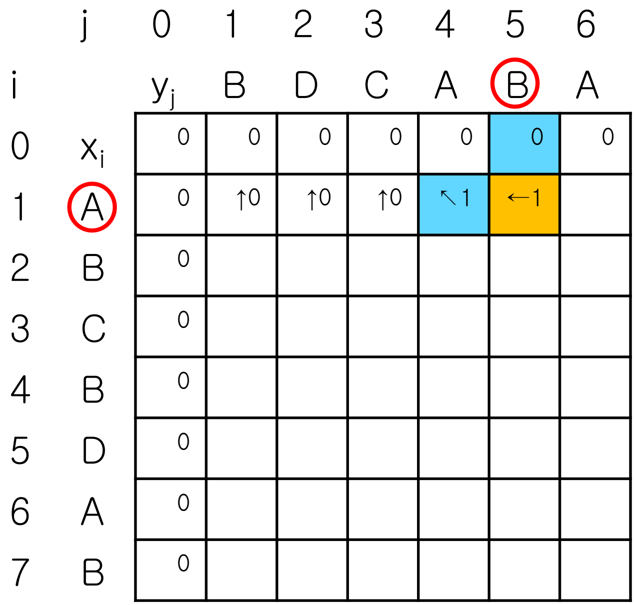
$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } \mathrm{x_i} = \mathrm{y_j} \\ \max(C[i,j-1], C[i-1,j]) & \text{if } i,j > 0 \text{ and } \mathrm{x_i} \neq \mathrm{y_j} \end{cases}$$

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[1,4] = C[0,3] + 1 = 1 \because \mathrm{x_1} = \mathrm{A} = \mathrm{A} = \mathrm{y_4}$

$B[1,4] = \text{"}\nwarrow\text{"} \because \mathrm{x_1} = \mathrm{A} = \mathrm{A} = \mathrm{y_4}$

Computing the Length of LCS



	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	↖1	←1	
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[1,5] = \max(C[1,4], C[0,5]) = 1 \ \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{B} = \mathrm{y_5}$

$B[1,5] = \text{"}\leftarrow\text{"} \because \mathrm{x_1} = \mathrm{A} \neq \mathrm{B} = \mathrm{y_5} \text{ and } C[1,4] > C[0,5]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0						
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

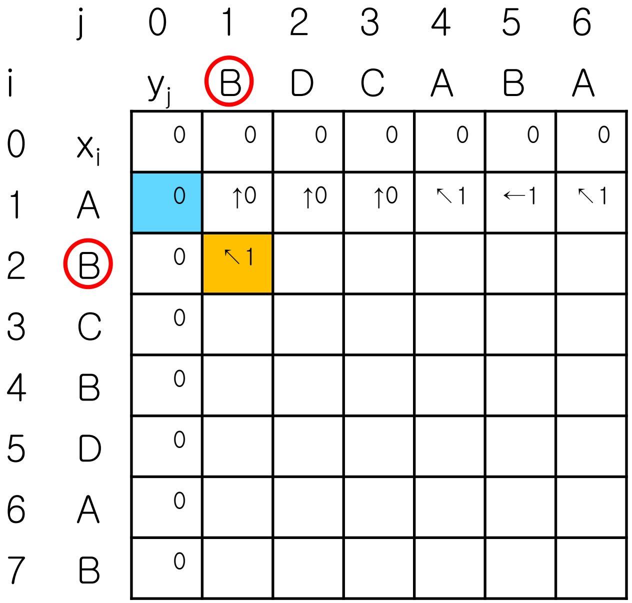
$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } \mathrm{x_i} = \mathrm{y_j} \\ \max(C[i,j-1], C[i-1,j]) & \text{if } i,j > 0 \text{ and } \mathrm{x_i} \neq \mathrm{y_j} \end{cases}$$

$$B[i,j] = \begin{cases} "\nwarrow" & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ "\uparrow" & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ "\leftarrow" & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[1,6] = C[0,5] + 1 = 1 \because \mathrm{x_1} = \mathrm{A} = \mathrm{A} = \mathrm{y_6}$

$B[1,6] = "\nwarrow" \because \mathrm{x_1} = \mathrm{A} = \mathrm{A} = \mathrm{y_6}$

Computing the Length of LCS



	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	↖1	←1	↖1
2	B	0	↖1					
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[2,1] = C[1,0] + 1 = 1 \because \mathrm{x_2} = \mathrm{B} = \mathrm{B} = \mathrm{y_1}$$

$$B[2,1] = \text{"}\nwarrow\text{"} \because \mathrm{x_2} = \mathrm{B} = \mathrm{B} = \mathrm{y_1}$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1				
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[2,2] = \max(C[2,1], C[1,2]) = 1 \ \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{D} = \mathrm{y_2}$

$B[2,2] = \text{"}\leftarrow\text{"} \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{D} = \mathrm{y_2} \text{ and } C[2,1] > C[1,2]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1			
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[2,3] = \max(C[2,2], C[1,3]) = 1 \ \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{C} = \mathrm{y_3}$

$B[2,3] = \text{"}\leftarrow\text{"} \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{C} = \mathrm{y_3} \text{ and } C[2,2] > C[1,3]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1		
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[2,4] = \max(C[2,3], C[1,4]) = 1 \ \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_4}$$

$$B[2,4] = \text{"}\uparrow\text{"} \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_4} \text{ and } C[1,4] \geq C[2,3]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[2,5] = C[1,4] + 1 = 2 \quad \because \mathrm{x_2} = \mathrm{B} = \mathrm{B} = \mathrm{y_5}$

$B[2,5] = \text{"}\nwarrow\text{"} \because \mathrm{x_2} = \mathrm{B} = \mathrm{B} = \mathrm{y_5}$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0						
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[2,6] = \max(C[2,5], C[1,6]) = 2 \;\because \mathrm{x_2} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_6}$

$B[2,6] = \text{"}\leftarrow\text{"} \because \mathrm{x_2} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_6} \text{ and } C[2,5] > C[1,6]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1					
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"↖"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"↑"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"←"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[3,1] = \max(C[3,0], C[2,1]) = 1 \ \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{B} = \mathrm{y_1}$$

$$B[3,1] = \text{"↑"} \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{B} = \mathrm{y_1} \text{ and } C[2,1] \geq C[3,0]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1				
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[3,2] = \max(C[3,1], C[2,2]) = 1 \ \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{D} = \mathrm{y_2}$$

$$B[3,2] = \text{"}\uparrow\text{"} \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{D} = \mathrm{y_2} \text{ and } C[2,2] \geq C[3,1]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2			
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[3,3] = C[2,2] + 1 = 2 \quad \because \mathrm{x_3} = \mathrm{C} = \mathrm{C} = \mathrm{y_3}$

$B[3,3] = \text{"}\nwarrow\text{"} \because\because \mathrm{x_3} = \mathrm{C} = \mathrm{C} = \mathrm{y_3}$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2		
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$C[3,4] = \max(C[3,3], C[2,4]) = 2 \ \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{A} = \mathrm{y_4}$

$B[3,4] = \text{"}\leftarrow\text{"} \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{A} = \mathrm{y_4} \text{ and } C[3,3] > C[2,4]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$$C[3,5] = \max(C[3,4], C[2,5]) = 2 \quad \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{B} = \mathrm{y_5}$$

$$B[3,5] = \text{"}\uparrow\text{"} \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{B} = \mathrm{y_5} \text{ and } C[2,5] \geq C[3,4]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0						
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[3,6] = \max(C[3,5], C[2,6]) = 2 \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{A} = \mathrm{y_6}$

$B[3,6] = \text{"}\uparrow\text{"} \because \mathrm{x_3} = \mathrm{C} \neq \mathrm{A} = \mathrm{y_6}$ and $C[2,6] \geq C[3,5]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1					
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[4,1] = C[3,0] + 1 = 1 \because \mathrm{x_4} = \mathrm{B} = \mathrm{B} = \mathrm{y_1}$

$B[3,6] = \text{"}\nwarrow\text{"} \because \mathrm{x_4} = \mathrm{B} = \mathrm{B} = \mathrm{y_1}$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1				
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[4,2] = \max(C[4,1], C[3,2]) = 1 \quad \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{D} = \mathrm{y_2}$$

$$B[4,2] = \text{"}\uparrow\text{"} \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{D} = \mathrm{y_2} \text{ and } C[3,2] \geq C[4,1]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2			
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[4,3] = \max(C[4,2], C[3,3]) = 2 \ \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{C} = \mathrm{y_3}$

$B[4,3] = \text{"}\uparrow\text{"} \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{C} = \mathrm{y_3} \text{ and } C[3,3] \geq C[4,2]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2		
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[4,4] = \max(C[3,5], C[2,6]) = 2 \ \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_4}$

$B[4,4] = \text{"}\uparrow\text{"} \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_4} \text{ and } C[3,4] \geq C[4,3]$

Computing the Length of LCS

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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[4,5] = C[3,4] + 1 = 3 \because \mathrm{x_4} = \mathrm{B} = \mathrm{B} = \mathrm{y_5}$$

$$B[4,5] = \text{"}\nwarrow\text{"} \because \mathrm{x_4} = \mathrm{B} = \mathrm{B} = \mathrm{y_5}$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0						
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[4,6] = \max(C[4,5], C[3,6]) = 3 \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_6}$

$B[4,6] = \text{"}\leftarrow\text{"} \because \mathrm{x_4} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_6} \text{ and } C[4,5] > C[3,6]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1					
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[5,1] = \max(C[5,0], C[4,1]) = 1 \ \because \mathrm{x_5} = \mathrm{D} \neq \mathrm{B} = \mathrm{y_1}$$

$$B[5,1] = \text{"}\uparrow\text{"} \because \mathrm{x_5} = \mathrm{D} \neq \mathrm{B} = \mathrm{y_1} \text{ and } C[4,1] \geq C[5,0]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2				
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[5,2] = C[4,1]+1=2 \because \mathrm{x_5} = \mathrm{D} = \mathrm{D} = \mathrm{y_2}$

$B[5,2] = \text{"}\nwarrow\text{"} \because \mathrm{x_5} = \mathrm{D} = \mathrm{D} = \mathrm{y_2}$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2			
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[5,3] = \max(C[5,2], C[4,3]) = 2 \ \because \mathrm{x_5} = \mathrm{D} \neq \mathrm{C} = \mathrm{y_3}$

$B[5,3] = \text{"}\uparrow\text{"} \because \mathrm{x_5} = \mathrm{D} \neq \mathrm{C} = \mathrm{y_3} \text{ and } C[4,3] \geq C[5,2]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2		
6	A	0						
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } x_i = y_j \\ \text{"}\uparrow\text{"} & \text{if } x_i \neq y_j \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } x_i \neq y_j \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[5,4] = \max(C[5,3], C[4,4]) = 2 \quad \because x_5 = D \neq A = y_4$$

$$B[5,4] = \text{"}\uparrow\text{"} \because x_5 = D \neq A = y_4 \text{ and } C[4,4] \geq C[5,3]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	
6	A	0						
7	B	0						

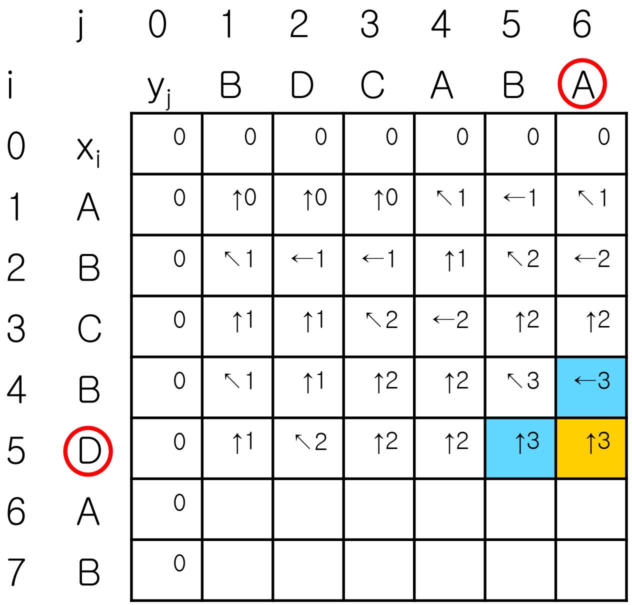
$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } \mathrm{x_i} = \mathrm{y_j} \\ \max(C[i, j-1], C[i-1, j]) & \text{if } i, j > 0 \text{ and } \mathrm{x_i} \neq \mathrm{y_j} \end{cases}$$

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$C[5,5] = \max(C[5,4], C[4,5]) = 3 \ \because \mathrm{x_5} = \mathrm{D} \neq \mathrm{B} = \mathrm{y_5}$

$B[5,5] = \text{"}\uparrow\text{"} \because \mathrm{x_5} = \mathrm{D} \neq \mathrm{B} = \mathrm{y_5} \text{ and } C[4,5] \geq C[5,4]$

Computing the Length of LCS



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0						
7	B	0						

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

$$B[i,j]=\begin{cases}\text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i}=\mathrm{y_j}\\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i}\neq\mathrm{y_j} \text{ and } C[i-1,j]\geq C[i,j-1]\\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i}\neq\mathrm{y_j} \text{ and } C[i,j-1]>C[i-1,j]\end{cases}$$

$C[5,6]=\max(C[5,5],C[4,6])=3 \because \mathrm{x_5}=\mathrm{D}\neq\mathrm{A}=\mathrm{y_6}$

$B[5,6]=\text{"}\uparrow\text{"} \because \mathrm{x_5}=\mathrm{D}\neq\mathrm{A}=\mathrm{y_6} \text{ and } C[4,6]\geq C[5,5]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1					
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } x_i = y_j \\ \text{"}\uparrow\text{"} & \text{if } x_i \neq y_j \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } x_i \neq y_j \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[6,1] = \max(C[6,0], C[5,1]) = 1 \because x_6 = \text{A} \neq \text{B} = y_1$

$B[6,1] = \text{"}\uparrow\text{"} \because x_6 = \text{A} \neq \text{B} = y_1 \text{ and } C[5,1] \geq C[6,0]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2				
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } x_i = y_j \\ \text{"}\uparrow\text{"} & \text{if } x_i \neq y_j \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } x_i \neq y_j \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[6,2] = \max(C[6,1], C[6,1]) = 2 \ \because x_6 = A \neq D = y_2$$

$$B[6,2] = \text{"}\uparrow\text{"} \because x_6 = A \neq D = y_2 \text{ and } C[5,2] \geq C[6,1]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2			
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$C[6,3] = \max(C[6,2], C[5,3]) = 2 \ \because \mathrm{x_6} = \mathrm{A} \neq \mathrm{C} = \mathrm{y_3}$

$B[6,3] = \text{"}\uparrow\text{"} \because \mathrm{x_6} = \mathrm{A} \neq \mathrm{C} = \mathrm{y_3} \text{ and } C[5,3] \geq C[6,2]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3		
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$C[6,4] = \mathrm{C}[5,3]+1=3 \because \mathrm{x}_6 = \mathrm{A} = \mathrm{A} = \mathrm{y}_4$

$B[6,4] = \text{"}\nwarrow\text{"} \because \mathrm{x}_6 = \mathrm{A} = \mathrm{A} = \mathrm{y}_4$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	
7	B	0						

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[6,5] = \max(C[6,4], C[5,5]) = 3 \because \mathrm{x_6} = \mathrm{A} \neq \mathrm{B} = \mathrm{y_5}$$

$$B[6,5] = \text{"}\uparrow\text{"} \because \mathrm{x_6} = \mathrm{A} \neq \mathrm{B} = \mathrm{y_5} \text{ and } C[5,5] \geq C[6,4]$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0						

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

$$B[i,j] = \begin{cases} "\nwarrow" & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ "\uparrow" & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ "\leftarrow" & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[6,6] = C[5,5] + 1 \because \mathrm{x_6} = \mathrm{A} = \mathrm{A} = \mathrm{y_6}$$

$$B[6,6] = "\nwarrow" \because \mathrm{x_6} = \mathrm{A} = \mathrm{A} = \mathrm{y_6}$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1					

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$$C[7,1] = C[6,0] + 1 = 1 \ \because \mathrm{x_7} = \mathrm{B} = \mathrm{B} = \mathrm{y_1}$$

$$B[7,1] = \text{"}\nwarrow\text{"} \because \mathrm{x_7} = \mathrm{B} = \mathrm{B} = \mathrm{y_1}$$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2				

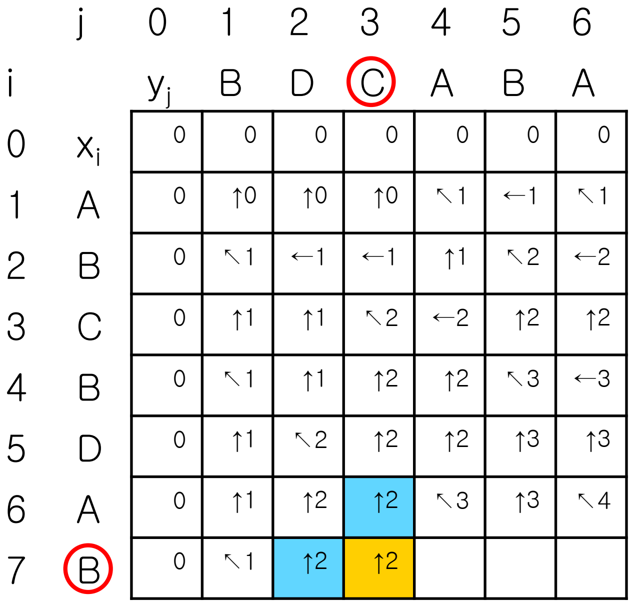
$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1,j-1]+1 & \text{if } i,j > 0 \text{ and } \mathrm{x_i} = \mathrm{y_j} \\ \max(C[i,j-1], C[i-1,j]) & \text{if } i,j > 0 \text{ and } \mathrm{x_i} \neq \mathrm{y_j} \end{cases}$$

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[7,2] = \max(C[7,1], C[7,2]) = 2 \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{D} = \mathrm{y_2}$

$B[7,2] = \text{"}\uparrow\text{"} \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{D} = \mathrm{y_2} \text{ and } C[6,2] \geq C[7,1]$

Computing the Length of LCS



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2			

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$C[7,3] = \max(C[7,2], C[6,3]) = 2 \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{C} = \mathrm{y_3}$

$B[7,3] = \text{"}\uparrow\text{"} \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{C} = \mathrm{y_3} \text{ and } C[6,3] \geq C[7,2]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3		

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[7,4] = \max(C[7,3], C[6,4]) = 3 \ \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_4}$

$B[7,4] = \text{"}\uparrow\text{"} \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_4} \text{ and } C[6,4] \geq C[7,3]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	

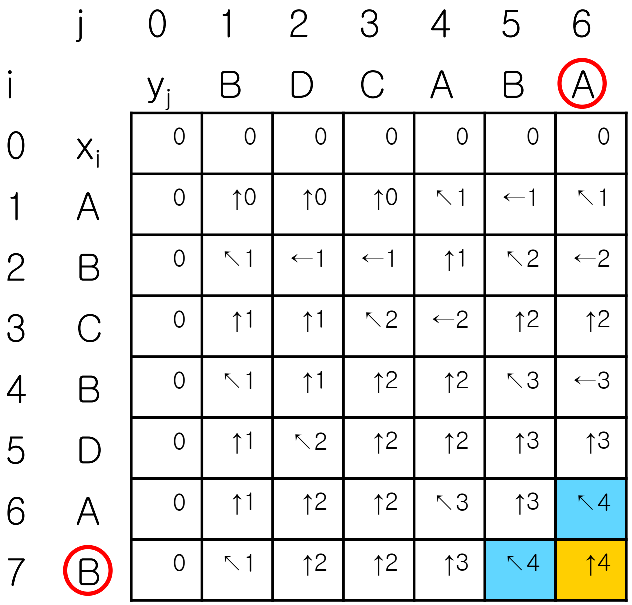
$$C[i,j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0 \\ C[i-1, j-1] + 1 & \text{if } i,j > 0 \text{ and } \mathrm{x_i} = \mathrm{y_j} \\ \max(C[i, j-1], C[i-1, j]) & \text{if } i,j > 0 \text{ and } \mathrm{x_i} \neq \mathrm{y_j} \end{cases}$$

$$B[i,j] = \begin{cases} \text{"↖"} & \text{if } \mathrm{x_i} = \mathrm{y_j} \\ \text{"↑"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1, j] \geq C[i, j-1] \\ \text{"←"} & \text{if } \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i, j-1] > C[i-1, j] \end{cases}$$

$C[7,5] = C[6,4] + 1 = 4 \quad \because \mathrm{x_7} = \mathrm{B} = \mathrm{B} = \mathrm{y_5}$

$B[7,5] = \text{"↖"} \because \mathrm{x_7} = \mathrm{B} = \mathrm{B} = \mathrm{y_5}$

Computing the Length of LCS



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

$C[7,6] = \max(C[7,5], C[6,6]) = 4 \ \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_6}$

$B[7,6] = \text{"}\uparrow\text{"} \because \mathrm{x_7} = \mathrm{B} \neq \mathrm{A} = \mathrm{y_6}$ and $C[6,6] \geq C[7,5]$

Computing the Length of LCS

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

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

$$B[i,j] = \begin{cases} \text{"}\nwarrow\text{"} & \text{if } \ \mathrm{x_i} = \mathrm{y_j} \\ \text{"}\uparrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i-1,j] \geq C[i,j-1] \\ \text{"}\leftarrow\text{"} & \text{if } \ \mathrm{x_i} \neq \mathrm{y_j} \text{ and } C[i,j-1] > C[i-1,j] \end{cases}$$

Constructing an LCS

- The b table returned by LCS-LENGTH enables us to quickly construct an LCS of $X=<x_1,x_2,\ldots,x_m>$ and $Y=<y_1, y_2,\ldots,y_n>$.
- We simply begin at b[m,n] and trace through the table by following the arrows. Whenever we encounter a "↖" in entry b[i, j], it implies that $x_i=y_j$ is an element of the LCS that LCS-LENGTH found.
- With this method, we encounter the elements of this LCS in reverse order.

Constructing an LCS

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

- **PRINT-LCS(b, X, X.length, Y.length)** is the initial invocation.
- Time Complexity : O(m+n)

Constructing an LCS

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

- PRINT-LCS(b,X,7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:7
j:6
Output:

- PRINT-LCS(b,X,7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:7
j:6
Output:

- PRINT-LCS(b,X,7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

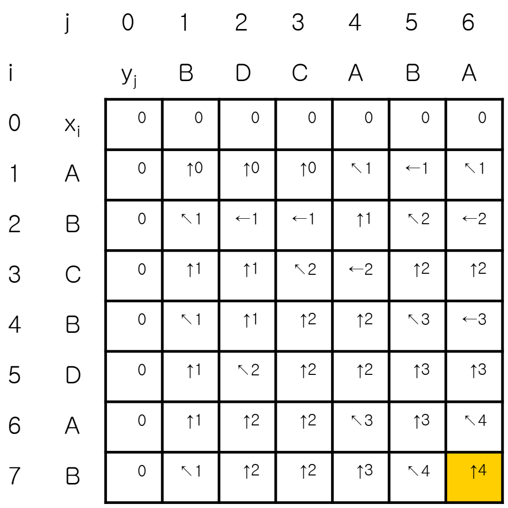
(i,j) = (7,6)

Constructing an LCS

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

i:7
j:6
Output:

- PRINT-LCS(b,X,7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

(i,j) = (7,6)

Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else    ⟶ PRINT-LCS(b,X,6,6)
        PRINT-LCS(b, X, i, j-1)
```

i:7
j:6
Output:

- PRINT-LCS(b,X,7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

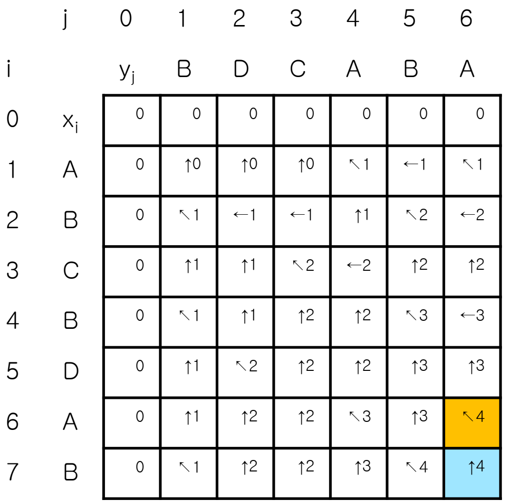
(i,j) = (7,6)

Constructing an LCS

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

i:6
j:6
Output:

- PRINT-LCS(b,X,7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

(i,j) = (6,6)
(i,j) = (7,6)

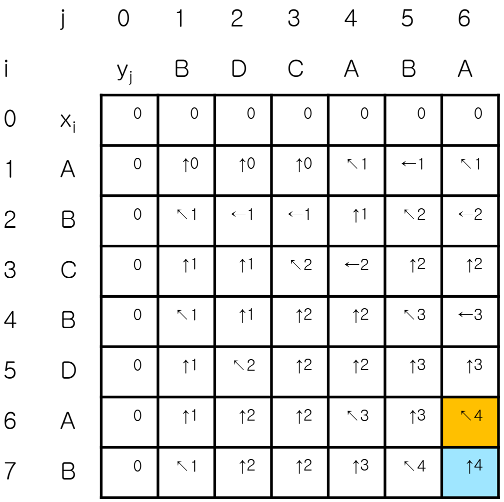
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print $x_i$  ⟶ PRINT-LCS(b,X,5,5)
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:6
j:6
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (6,6)
(i,j) = (7,6)



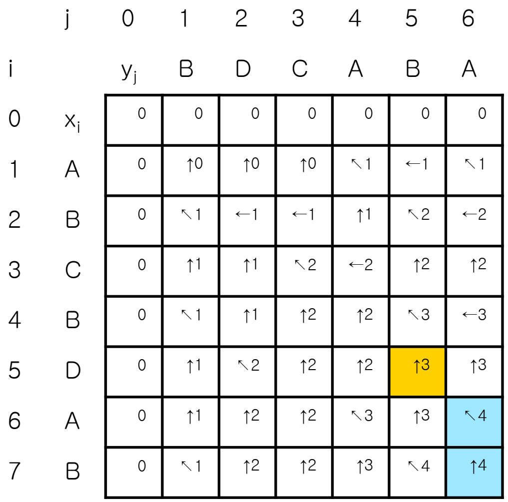
	j	0	1	2	3	4	5	6
i		y_j	B	D	C	A	B	A
0	x_i	0	0	0	0	0	0	0
1	A	0	↑0	↑0	↑0	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:5
j:5
Output:

- PRINT-LCS(b,X,7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

Constructing an LCS

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

i:5
j:5
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

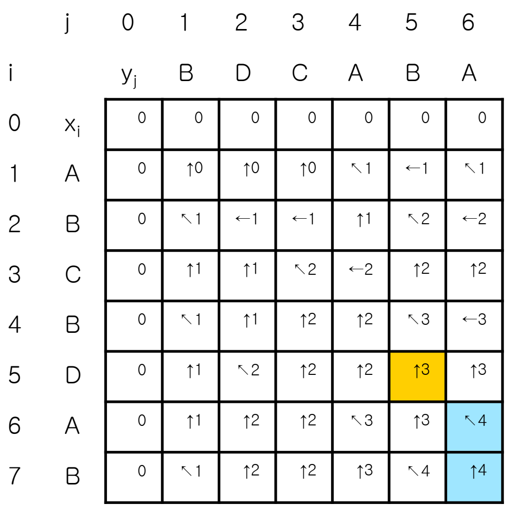
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print $x_i$
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else    └──→ PRINT-LCS(b,X,4,5)
        PRINT-LCS(b, X, i, j-1)
```

i:5
j:5
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

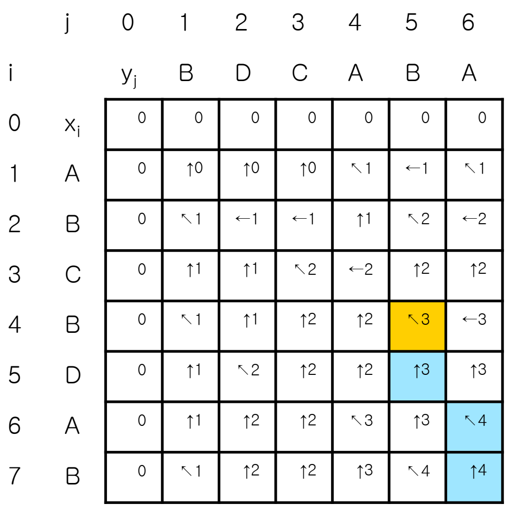
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖ "
        PRINT-LCS(b, X, i-1, j-1)
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:4
j:5
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

→ PRINT-LCS(b,X,3,4)

i:4
j:5
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

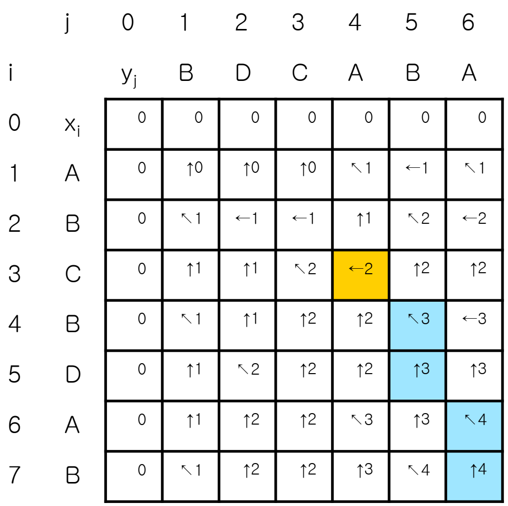
Constructing an LCS

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

i:3
j:4
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



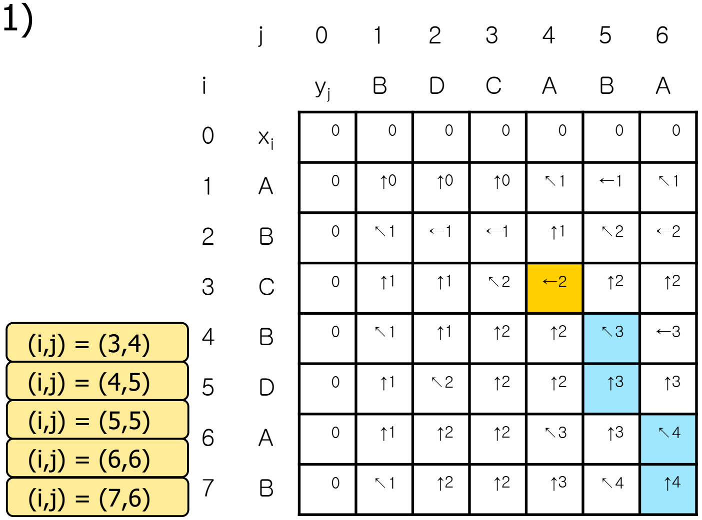
	j	0	1	2	3	4	5	6
i		y_j	B	D	C	A	B	A
0	x_i	0	0	0	0	0	0	0
1	A	0	↑0	↑0	↑0	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:3
j:4
Output:

- PRINT-LCS(b,X,7,6)



(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

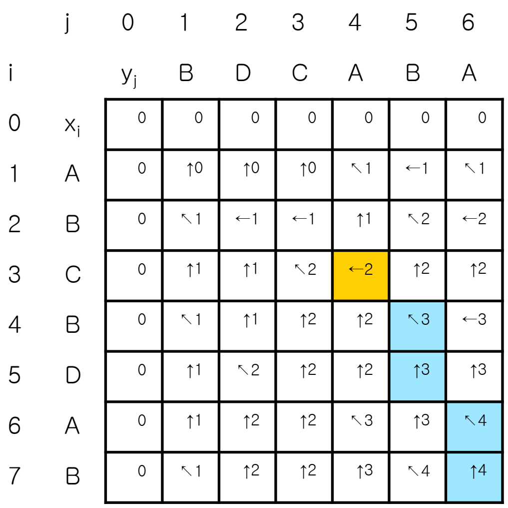
Constructing an LCS

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

i:3
j:4
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

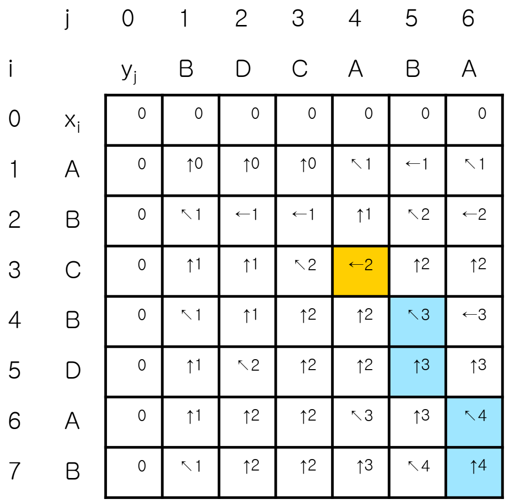
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print $x_i$
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
          → PRINT-LCS(b,X,3,3)
```

i:3
j:4
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

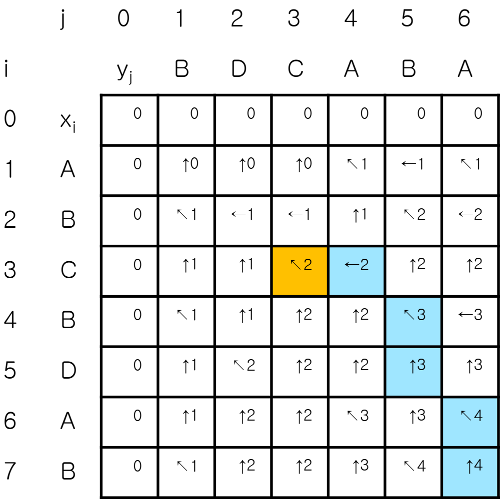
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:3
j:3
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

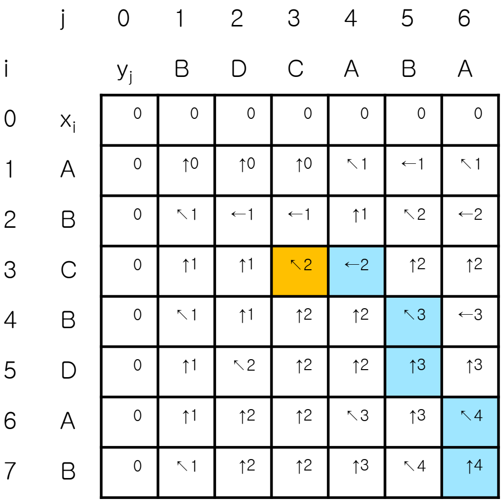
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖ "
        PRINT-LCS(b, X, i-1, j-1)
        print x_i  ⟶ PRINT-LCS(b,X,2,2)
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:3
j:3
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

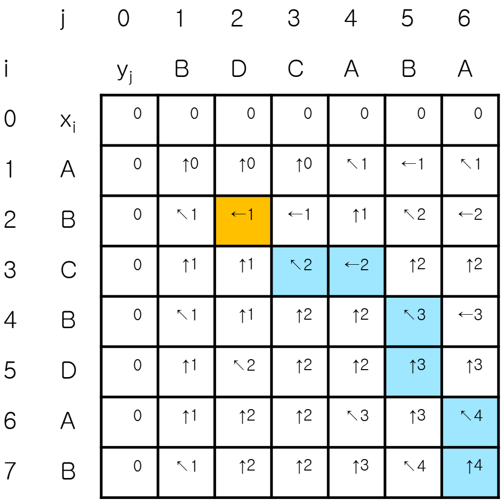
Constructing an LCS

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

i:2
j:2
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

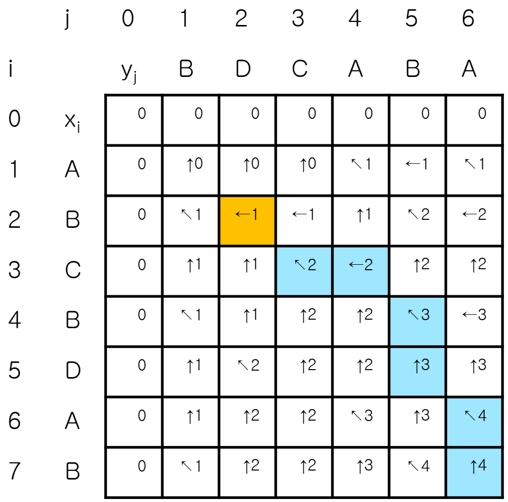
Constructing an LCS

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

i:2
j:2
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

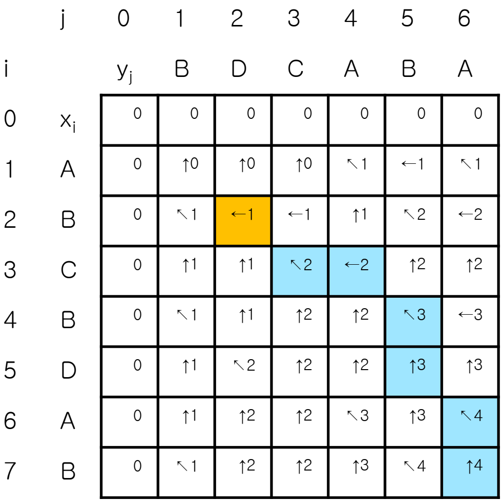
Constructing an LCS

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

i:2
j:2
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

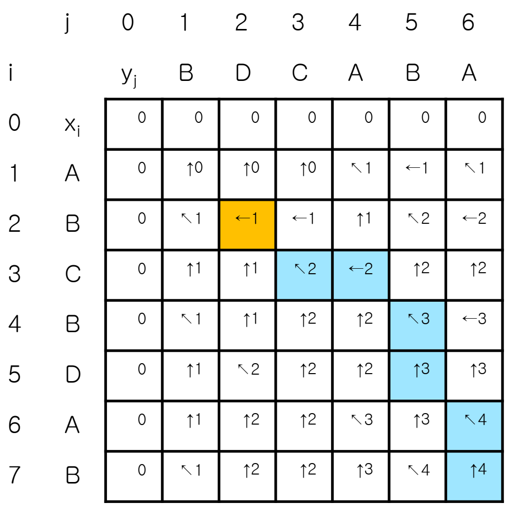
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print $x_i$
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
          →PRINT-LCS(b,X,2,1)
```

i:2
j:2
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

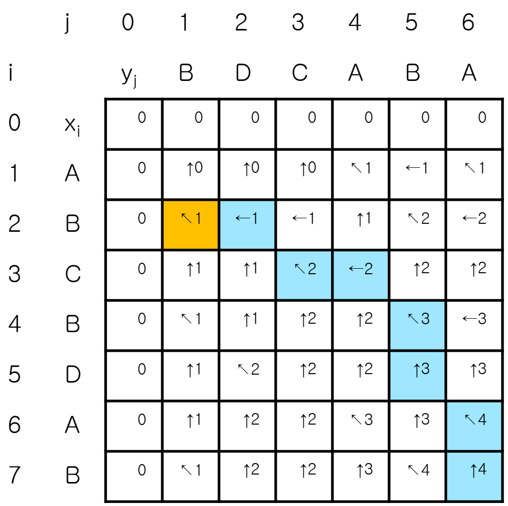
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖ "
        PRINT-LCS(b, X, i-1, j-1)
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:2
j:1
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (2,1)
(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

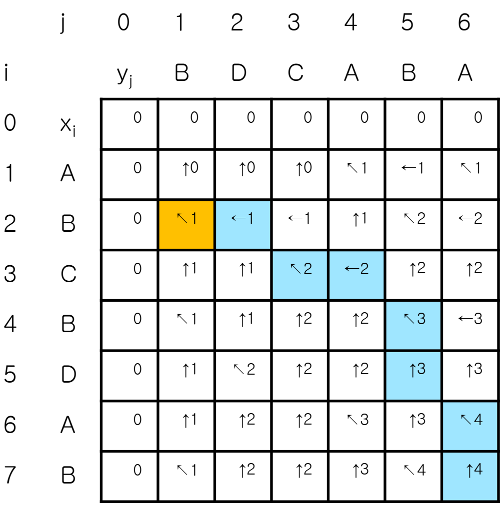
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖ "
        PRINT-LCS(b, X, i-1, j-1)
        print x_i  ⟶ PRINT-LCS(b,X,1,0)
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:2
j:1
Output:

- PRINT-LCS(b,X,7,6)

(i,j) = (2,1)
(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



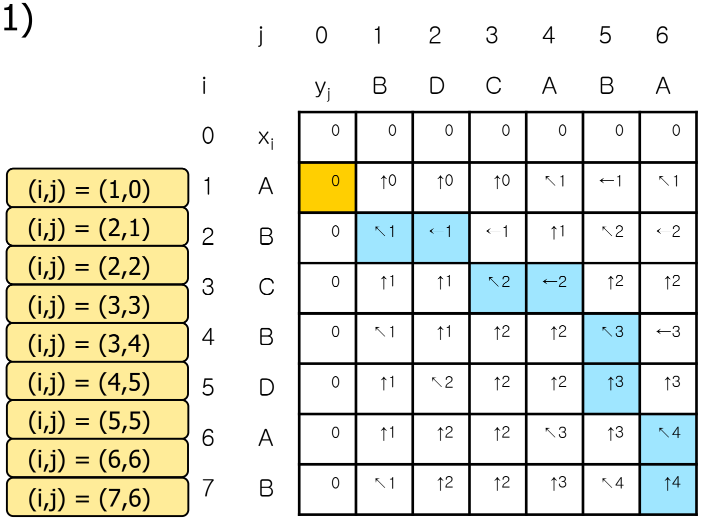
	j	0	1	2	3	4	5	6
i		y_j	B	D	C	A	B	A
0	x_i	0	0	0	0	0	0	0
1	A	0	↑0	↑0	↑0	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:1
j:0
Output:

- PRINT-LCS(b,X,7,6)



(i,j) = (1,0)
(i,j) = (2,1)
(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

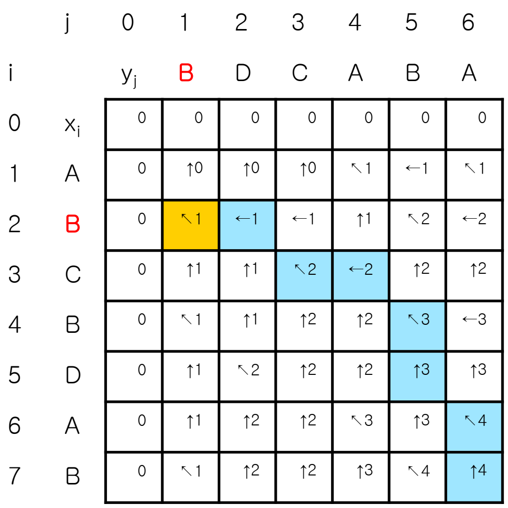
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖ "
        PRINT-LCS(b, X, i-1, j-1)  return
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:2
j:1
Output: B

- PRINT-LCS(b,X,7,6)

(i,j) = (2,1)
(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:2
j:2
Output: B

return

(i,j) = (2,2)
(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

- PRINT-LCS(b,X,7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

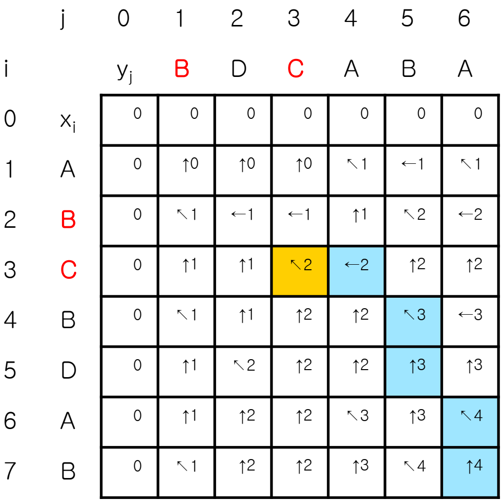
Constructing an LCS

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

i:3
j:3
Output: BC

- PRINT-LCS(b,X,7,6)

(i,j) = (3,3)
(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

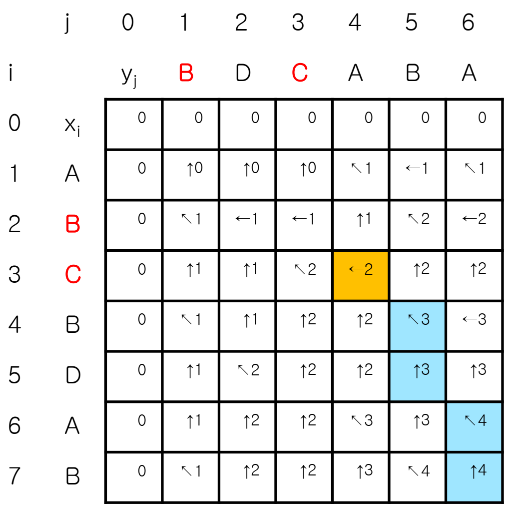
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
1.  if i==0 or j==0
2.      return
3.  if b[i, j] == "↖ "
4.      PRINT-LCS(b, X, i-1, j-1)
5.      print $x_i$
6.  else if b[i, j] == "↑"
7.      PRINT-LCS(b, X, i-1, j)
8.  else
9.      PRINT-LCS(b, X, i, j-1)  return
```

i:3
j:4
Output: BC

- PRINT-LCS(b,X,7,6)

(i,j) = (3,4)
(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖ "
        PRINT-LCS(b, X, i-1, j-1)  return
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)
    else
        PRINT-LCS(b, X, i, j-1)
```

i:4
j:5
Output: BCB

- PRINT-LCS(b,X,7,6)

(i,j) = (4,5)
(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

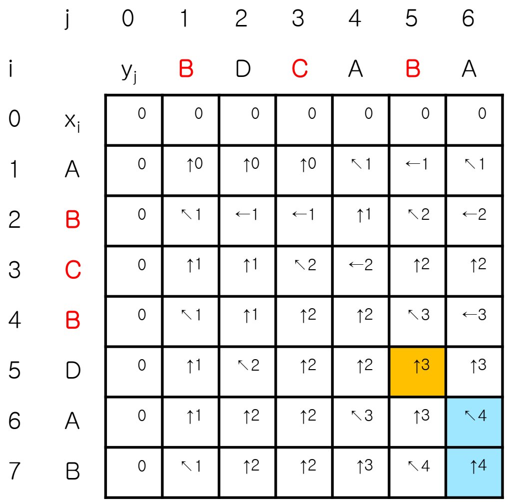
Constructing an LCS

```
PRINT-LCS(b, X, i, j)
    if i==0 or j==0
        return
    if b[i, j] == "↖"
        PRINT-LCS(b, X, i-1, j-1)
        print x_i
    else if b[i, j] == "↑"
        PRINT-LCS(b, X, i-1, j)     return
    else
        PRINT-LCS(b, X, i, j-1)
```

i:5
j:5
Output: BCB

- PRINT-LCS(b,X,7,6)

(i,j) = (5,5)
(i,j) = (6,6)
(i,j) = (7,6)



	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:6
j:6
Output: BCBA

- PRINT-LCS(b,X,7,6)

(i,j) = (6,6)
(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:7
j:6
Output: BCBA

- PRINT-LCS(b,X,7,6)

(i,j) = (7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

i:7
j:6
Output: BCBA

- PRINT-LCS(b,X,7,6)

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Constructing an LCS

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

Output: BCBA

- PRINT-LCS(b,X,7,6)
    - BCBA is an optimal solution

	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	↖1	←1	↖1
2	B	0	↖1	←1	←1	↑1	↖2	←2
3	C	0	↑1	↑1	↖2	←2	↑2	↑2
4	B	0	↖1	↑1	↑2	↑2	↖3	←3
5	D	0	↑1	↖2	↑2	↑2	↑3	↑3
6	A	0	↑1	↑2	↑2	↖3	↑3	↖4
7	B	0	↖1	↑2	↑2	↑3	↖4	↑4

Improving the code

- We can eliminate the b table altogether.
- Each c[i,j] entry depends on only three other c table entries.
  - c[i-1,j-1], c[i-1,j], and c[i,j-1]
- Given the value of c[i,j], we can determine in O(1) time which of these three values was used to compute c[i,j], without inspecting table b.
- Thus, we can reconstruct an LCS in O(m+n) time using a procedure similar to PRINT-LCS.
- Although we save θ(mn) space by this method, the auxiliary space requirement for computing an LCS does not asymptotically decrease, since we need θ(mn) space for the c table anyway.

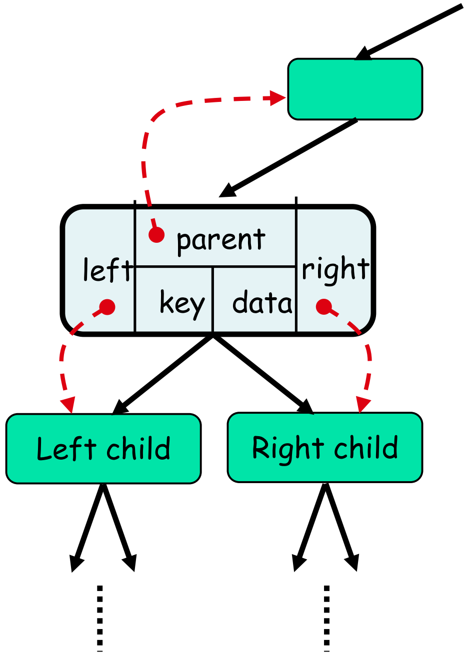
Improving the code

- We can, however, reduce the asymptotic space requirements for LCS-LENGTH, since it needs only two rows of table c at a time.
  - the row being computed and the previous row
- This improvement works if we need only the length of an LCS.
- If we need to reconstruct the elements of an LCS, the smaller table does not keep enough information to retrace our steps in O(m+n) time.

Optimal Binary Search Trees

A Binary Search Tree

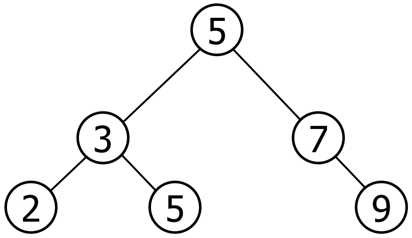
- Tree representation:
    - A linked data structure in which each node is an object.
- Node representation:
    - key field
    - (satellite) data
    - left: pointer to left child
    - right: pointer to right child
    - parent: pointer to parent
- It satisfies the binary-search-tree property!



Binary Search Tree Property

- If y is in left subtree of x,
  then y.key ≤ x.key

- If y is in right subtree of x,
  then y.key ≥ x.key

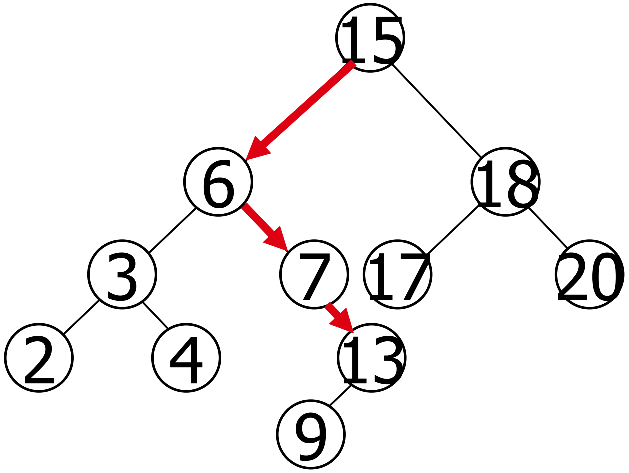


Searching for a Key

- Given a pointer to the root of a tree and a key k:
  - Return a pointer to a node with key k, if one exists
  - Otherwise return NIL
- Starting at the root: trace down a path by comparing k with the key of the current node:
  - If the keys are equal: we have found the key
  - If $k < x.key$ search in the left subtree of x
  - If $k > x.key$ search in the right subtree of x

Searching for a Key

- Search for key 13:
  - 15 → 6 → 7 → 13



Searching for a Key

```
TREE-SEARCH(x, k)
if x == NIL or k == x.key
    then return x
if k < x.key
    then return TREE-SEARCH(x.left, k )
    else return TREE-SEARCH(x.right, k )
```

- Running Time: $O(h)$, where h is the height of the tree.

Optimal Binary Search Trees

- Suppose that we are designing a program to translate text from English to French.
- For each occurrence of each English word in the text, we need to look up its French equivalent.
- We could perform these lookup operations by building a binary search tree with n English words as keys and their French equivalents as satellite data.
- Because we will search the tree for each individual word in the text, we want the total time spent searching to be as low as possible.

Optimal Binary Search Trees

- Words appear with different frequencies, however, and a frequently used word may appear far from the root while a rarely used word appears near the root.
- Such an organization would slow down the translation, since the number of nodes visited when searching for a key in a binary search tree equals one plus the depth of the node containing the key.
- We want words that occur frequently in the text to be placed nearer the root.
- Moreover, some words in the text might have no French translation, and such words would not appear in the binary search tree at all.

Optimal Binary Search Trees

- How do we organize a binary search tree so as to minimize the number of nodes visited in all searches, given that we know how often each word occurs?
- What we need is known as an optimal binary search tree.

Problem Formulation

- We are given a sequence K = <k_1, k_2, …, k_n> of n distinct keys in sorted order, and we wish to build a binary search tree from these keys.
- For each key k_i, we have a probability p_i that a search will be for k_i.
- Some searches may be for values not in K, and so we also have n + 1 “dummy keys” d_0, d_1, d_2,…, d_n representing values not in K.
- In particular,
    - d_0 represents all values less than than k_1
    - d_n represents all values greater than k_n
    - the dummy key d_i represents all values between k_i and k_{i+1}, for i = 1,2,…,n-1,
- For each dummy key d_i, we have a probability q_i that a search will correspond to d_i.

Why are Dummy Keys Necessary?

d_0		d_1		d_2		d_3		d_4		d_5
…	k_1	…	k_2	…	k_3	…	k_4	…	k_5	…

- For each key k_i, we have a probability p_i that a search will be for k_i.
- For each dummy key d_i, we have a probability q_i that a search will correspond to d_i.

What Effect Do Dummy Keys Have on the Problem?

- Each key k_i is an internal node, and each dummy key d_i is a leaf.
- Every search is either successful (finding some key k_i) or unsuccessful (finding some dummy key d_i), and so we have

$$\sum_{i=1}^{n} p_i + \sum_{i=0}^{n} q_i = 1$$

The Expected Cost of a Search in a Tree

- Let us assume that the actual cost of a search is the number of nodes examined, i.e., the depth of the node found by the search in T plus 1.
- Then, the expected cost of a search in *T* is

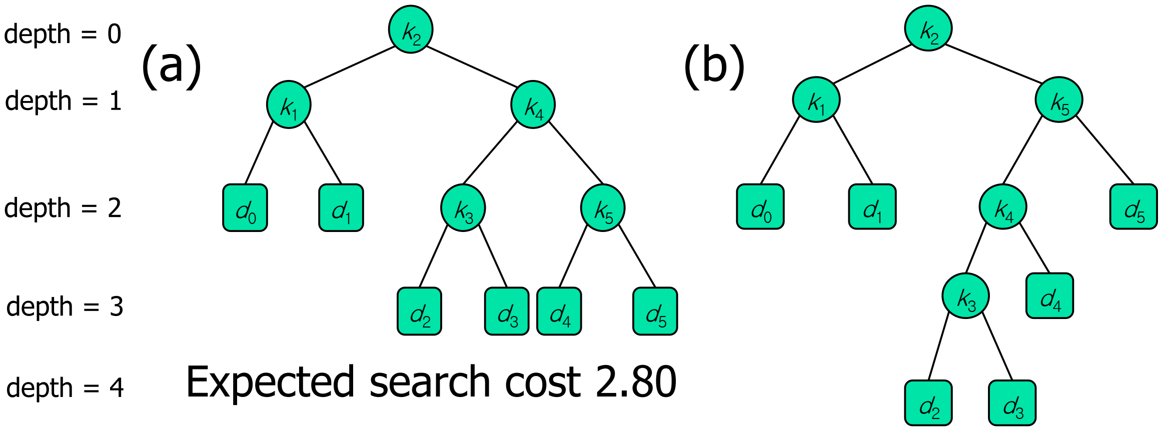
$$\text{E[search cost in } T\text{]} = \sum_{i=1}^{n} (depth_T(k_i)+1) \cdot p_i + \sum_{i=0}^{n} (depth_T(d_i)+1) \cdot q_i$$

$$= 1 + \sum_{i=1}^{n} depth_T(k_i) \cdot p_i + \sum_{i=0}^{n} depth_T(d_i) \cdot q_i$$

where depth$_T$ denotes a node's depth in the tree T.

- We wish to build a binary search tree such that the expected cost of a search in T is minimized.
- We call such a tree an optimal binary search tree.

An Example of Binary Search Trees



Expected search cost 2.80

Expected search cost 2.75

- Binary search trees for 5 keys with the following probabilities:

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

An Optimal Binary Search Tree

- An optimal binary search tree is not necessarily a tree whose overall height is smallest.
- Nor can we necessarily construct an optimal binary search tree by always putting the key with the greatest probability at the root.
- Here, key k_5 has the greatest search probability of any key, yet the root of the optimal binary search tree shown previously is k_2.
- The lowest expected cost of any binary search tree with k_5 at the root is 2.85.

The Exhaustive Search is Bad!

- As with matrix-chain multiplication, exhaustive checking of all possibilities fails to yield an efficient algorithm.
    - The number of binary trees with n nodes is $\Omega(4^n/n^{3/2})$.
- We will solve this problem with dynamic programming.

The Optimal Structure

- Observations about subtrees
  - Consider any subtree of a binary search tree.
  - It must contain keys in a contiguous range $k_i,\ldots,k_j$, for some $1 \le i \le j \le n$.
  - In addition, a subtree that contains keys $1 \le i \le j \le n$ must also have as its leaves the dummy keys $d_{i-1},\ldots,d_j$.

The Optimal Structure

- If an optimal binary search tree T has a subtree T′ containing keys $k_i,\ldots,k_j$, then T′ must be optimal as well for the subproblem with keys $k_i,\ldots,k_j$ and dummy keys $d_{i-1},\ldots,d_j$.
- The usual cut-and-paste argument applies. If there were a subtree T″ whose expected cost is lower than that of T′, then we could cut T′ out of T and paste in T″, resulting in a binary search tree of lower expected cost than T, thus contradicting the optimality of T.

The Structure of an Optimal Binary Search Tree

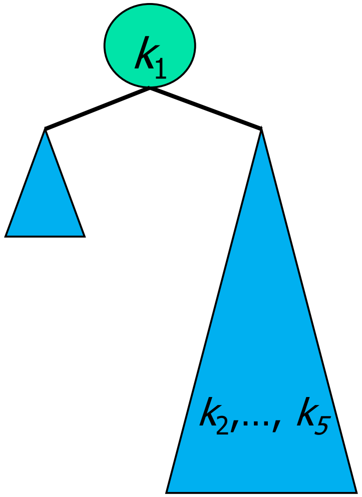
- Given keys k_i,..., k_j , one of these keys, say k_r ($i \le r \le j$), will be the root of an optimal subtree containing these keys.
- The left subtree of the root k_r will contain the keys k_i,..., k_{r-1} (and dummy keys d_{i-1},...,d_{r-1}).
- The right subtree will contain the keys k_{r+1}, ... k_j (and dummy keys d_r,... d_j).
- As long as we examine all the candidate roots k_r, where $i \le r \le j$, and we determine all optimal binary search trees containing k_i,...,k_{r-1} and those containing k_{r+1},...,k_j, we find an optimal binary search tree.

Empty Subtrees

- In a subtree with keys k_i,..., k_j, if we select k_i as the root, k_i 's left subtree contains the keys k_i,..., k_{i-1}.
- Symmetrically, if we select k_j as the root, k_j 's right subtree contains the keys k_{j+1},..., k_j.
- We adopt the convention that
  - A subtree containing keys k_i,..., k_{i-1} has no actual keys but does contain the single dummy key d_{i-1}.
  - A subtree containing keys k_{j+1},..., k_j has no actual keys but does contain the single dummy key d_j.

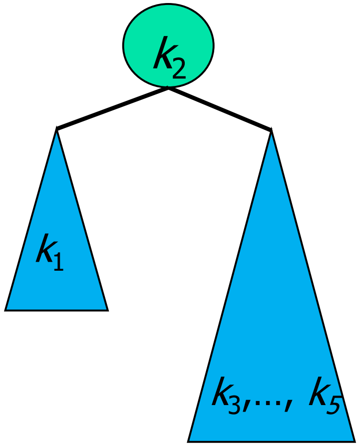
Dynamic Programming

- $K = \langle k_1, k_2, \ldots, k_5 \rangle$
- Root : k_1



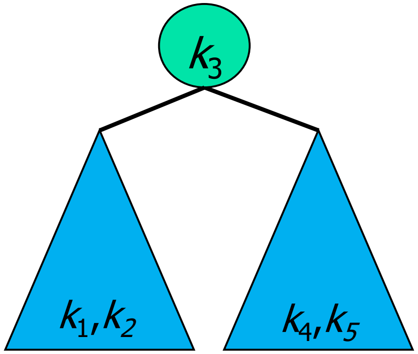
Dynamic Programming

- $K = <k_1, k_2, \ldots, k_5>$
- Root : k_2



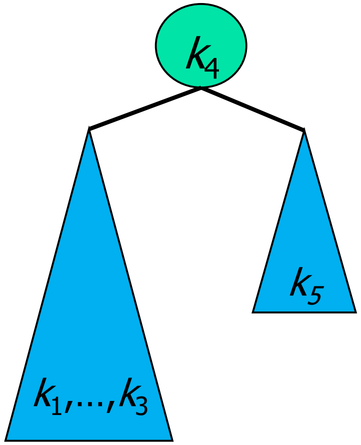
Dynamic Programming

- $K = <k_1, k_2, \ldots, k_5>$
- Root : k_3



Dynamic Programming

- $K = <k_1, k_2, \ldots, k_5>$
- Root : k_4



Dynamic Programming

- $K = <k_1, k_2, \ldots, k_5>$
- Root : k_5