# CS4311 Design and Analysis of Algorithms

Tutorial: Hirschberg's Trick for LCS

## Space for Finding LCS

- In the lecture, we see that LCS problem can be solved using O(mn) space
- And if we only need the length of LCS
  we can do so by only keeping current row
  and previous row → reduce space to O(n)
- Note: We can also fill L column by column
   Space usage: O(min { m,n } )

## Reducing Space Usage

Question: How about getting the LC5? Can we do so with O(n) space?

#### Solution I:

- Use O(mn) time to find the last row
- Use O(mn) time to find the 2<sup>nd</sup> last row
- •
- Use O(mn) time to find the first row
- $\rightarrow$  Total time:  $O(m^2n)$

Let  $S_1$  and  $S_2$  denote the first half and the second half of S, respectively

( $S_1$  and  $S_2$  have equal length)

Consider X, which is the LCS of S and T.

Let X' and X" denote the portion of X which comes from  $S_1$  and  $S_2$ 

(X' or X" may be empty, and may be of unequal length)

# Example

```
S = DIRTYROOM
T = DORMITORY
X = DITR
```

$$S_1 = DIRT$$
,  $S_2 = YROOM$   
 $X' = DIT$ ,  $X'' = R$ 

#### Observation:

If X' and X" come from  $T_{1,r}$  and  $T_{r+1,n}$  for some r, then

- X' is an LCS of  $S_1$  and  $T_{1,r}$
- X" is an LCS of  $S_2$  and  $T_{r+1,n}$

Corollary: The reverse of X" is LCS of the reverse of  $S_2$  and reverse of  $T_{r+1,n}$ 

```
Let len_{i,j} = length of the LCS of S_{1,i} and T_{1,j}

Let rev_{i,j} = length of the LCS of S_{i,m} and T_{j,n}

= length of the LCS of reverse of S_{i,m} and reverse of T_{j,n}
```

```
Lemma: len_{m,n} = max_r \{ len_{m/2,r} + rev_{m/2+1,r+1} \}
And, if r = r^* achieves the above max,
```

- X' is an LCS of  $S_1$  and  $T_{1,r^*}$
- X" is an LCS of  $S_2$  and  $T_{r^{*+1},n}$

Based on the previous lemma, we can find r\* as follows:

```
Step 1: Fill L for row 1 to row m/2

(from top-left corner)

Step 2: Fill L for row m to row m/2 + 1

(from bottom-right corner)

Step 3: Find r* from rows m/2 and m/2+1
```

	D	0	R	M	I	T	0	R	У	
									-	
D										
I										
R										
T										
У										
R										
0										
0										
M										

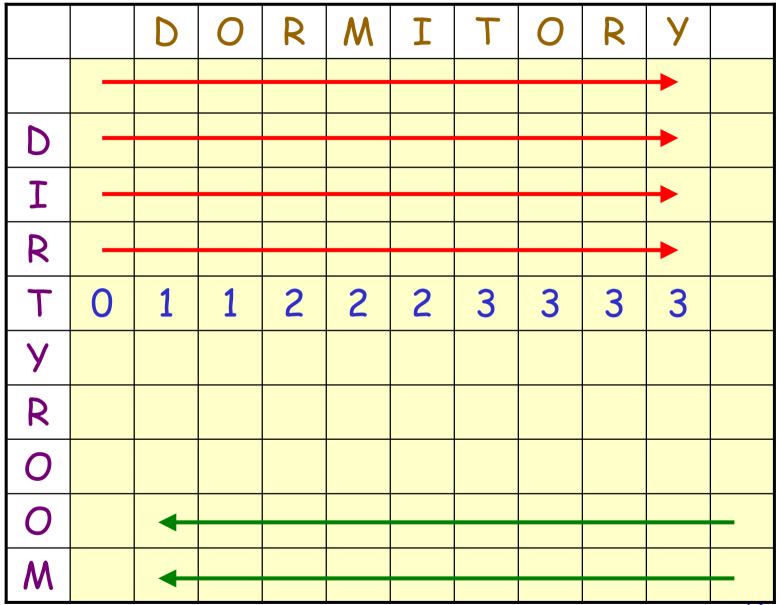
		D	0	R	M	I	T	0	R	У	
										-	
D	_									<b>→</b>	
I											
R											
Т											
У											
R											
0											
0											
M											

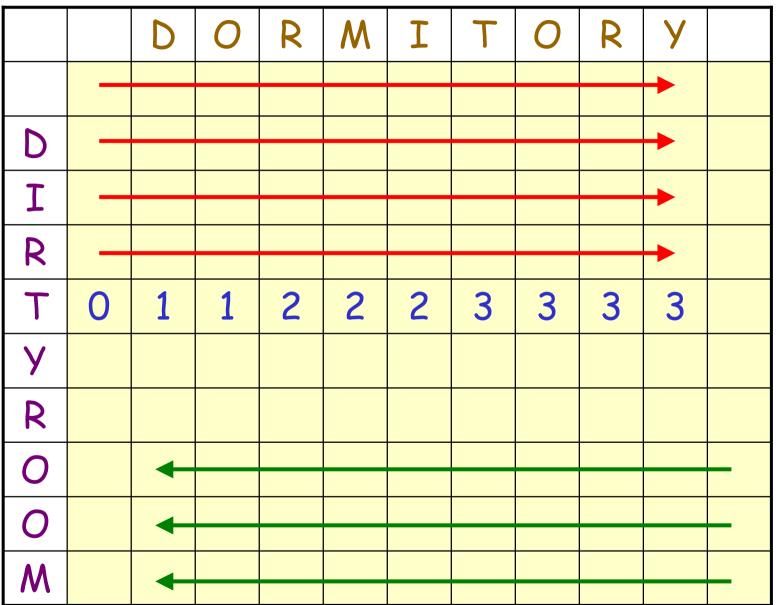
		D	0	R	M	I	T	0	R	У	
										-	
D	_									-	
I	_									-	
R											
T											
У											
R											
0											
0											
M											

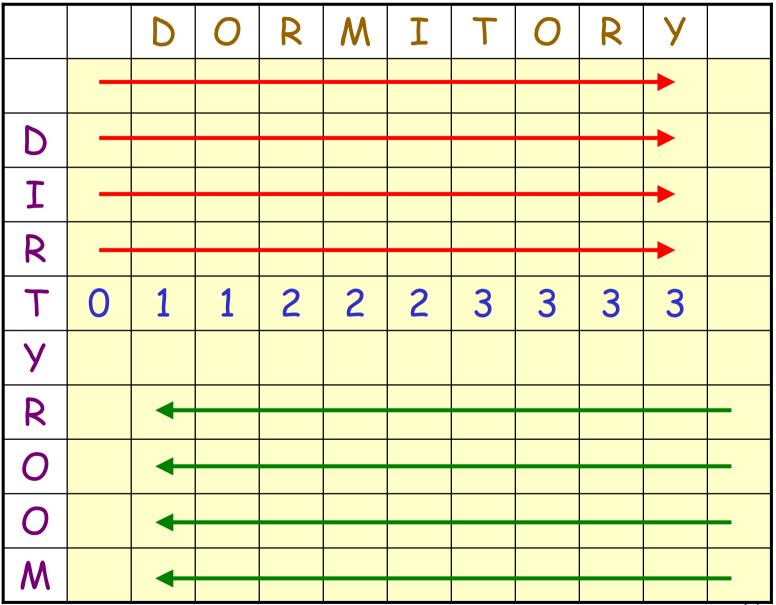
		D	0	R	M	I	T	0	R	У	
										-	
D	_									-	
I										-	
R	_									-	
Т											
У											
R											
0											
0											
M											

		D	0	R	M	I	T	0	R	У	
										<b>→</b>	
D										-	
I										<b>→</b>	
R										<b>→</b>	
Т	0	1	1	2	2	2	3	3	3	3	
У											
R											
0											
0											
M											_

		D	0	R	M	I	T	0	R	У	
	-									<b>→</b>	
D	_									-	
I	_									-	
R	_									-	
T	0	1	1	2	2	2	3	3	3	3	
У											
R											
0											
0											
M		+									







		D	0	R	M	I	T	0	R	У	
										<b>→</b>	
D										-	
I										-	
R										-	
Т	0	1	1	2	2	2	3	3	3	3	
У		2	2	2	1	1	1	1	1	1	0
R		<b>+</b>									_
0		<b>+</b>									_
0		<b>+</b>									_
M		<b>4</b>									_

#### Example Run: Step 3 (Find r\*)

		D	0	R	M	I	T	0	R	У	
D											
I											
R											
T	0	1	1	2	2	2	3	3	3	3	
У		2	2	2	1	1	1	1	1	1	0
R											
0											
0											
M											

- After finding r\*, we can recursively find
  - (i) LCS of  $S_{1,m/2}$  and  $T_{1,r*}$
  - (ii) LCS of  $S_{m/2+1,m}$  and  $T_{r^{*+1,n}}$
- Total Space: O(n) because space can be reused!
- · Total Time:

$$T(m,n) = T(m/2,r^*) + T(m/2, n-r^*) + \Theta(mn)$$

 $\rightarrow$  By recursion-tree,  $T(m,n) = \Theta(mn)$