



# DESIGN AND ANALYSIS OF ALGORITHMS

## UE19CS251

---

**Shylaja S S**

Department of Computer Science  
& Engineering

# DESIGN AND ANALYSIS OF ALGORITHMS

---

## String Matching

Major Slides Content: Anany Levitin

**Shylaja S S**

Department of Computer Science & Engineering

### String Matching - Terms

- pattern:  
a string of  $m$  characters to search for
- text:  
a (longer) string of  $n$  characters to search in
- problem:  
find a substring in the text that matches the pattern

Step 1: Align pattern at beginning of text

Step 2: Moving from left to right, compare each character of pattern to the corresponding character in text until:

- all characters are found to match (successful search); or
- a mismatch is detected

Step 3: While pattern is not found and the text is not yet exhausted, realign pattern one position to the right and repeat Step 2

```
ALGORITHM BruteForceStringMatch(T[0 .. n - 1], P[0 .. m - 1])
//Implements brute-force string matching
//Input: An array T[0 .. n - 1] of n characters representing a text
// and an array P[0 .. m - 1] of m characters representing a pattern
//Output: The index of the first character in the text that starts a
//matching substring or -1 if the search is unsuccessful
for i ← 0 to n-m do
    j ← 0
    while j < m and P[j] = T[i + j] do
        j ← j+1
    if j = m return i
return -1
```

# DESIGN AND ANALYSIS OF ALGORITHMS

## String Matching Example

T	H	I	S		I	S		A		S	I	M	P	L	E		E	X	A	M	P	L	E
S	I	M	P	L	E																		
	S	I	M	P	L	E																	
		S	I	M	P	L	E																
			S	I	M	P	L	E															
				S	I	M	P	L	E														
					S	I	M	P	L	E													
						S	I	M	P	L	E												
							S	I	M	P	L	E											
								S	I	M	P	L	E										
									S	I	M	P	L	E									
										S	I	M	P	L	E								
											S	I	M	P	L	E							

Worst Case:

- The algorithm might have to make all the 'm' comparisons for each of the  $(n-m+1)$  tries
- Therefore, the algorithm makes  $m(n-m+1)$  comparisons
- Brute Force String Matching is a  $O(nm)$  algorithm



# THANK YOU

---

**Shylaja S S**

Department of Computer Science  
& Engineering

**[shylaja.sharath@pes.edu](mailto:shylaja.sharath@pes.edu)**