

String Matching (Naïve Algorithm: Brute Force)

- Given pattern P : String (length: n)
and text T : String (length: m)
Find one or all occurrences of P in T .

Pattern ANT

Text IAMANANTELOPE

I A M A N A N T E L O P E

A N T

A N T

A N T

A N T

A N T

A N T

Analysis

- Worst case example:
 - aaaaa.....b input text (length n)
 - aa..b pattern (length m)
- Mismatch $(n - m)$ times
- Success at $(n - m) + 1$
- Total Comparisons $((n - m) + 1) \times m$
- $\Theta(nm)$

String Matching (Horspool's Algorithm)

- Make "Bad Match Table"
- Compare pattern to text, starting from rightmost character in the pattern
 - If mismatch, move pattern forward corresponding to value in the table (8)

analysis



Pattern TOOTH

Text TRUSTHARDTOOTHBRUSHES

T R U S T T H A R D T O O T H B R U S H E S

TOOT H

TOOTH

TOOTH

TOOTH

0 1 2 3 4
T O O T H ; length = 5 default

Letter	T	O	H	
Value	4	2	5	5

$T = 5 - 0 - 1 = 4$
 $O = 5 - 1 - 1 = 3$
 $O = 5 - 2 - 1 = 2$
 $T = 5 - 3 - 1 = 1$
 $H = 5$ (last letter = length; if not already defined)

value = length - index - 1

- Worst case same as naïve example:
 - 1st input text (length n)
 - 0111...1 pattern (length m)
- Worst Case $O(nm)$
- Best case
 - 1st input text (length n)
 - 0^m pattern (length m)
- Best case $O(m/n)$
- Average case $O(m/|Σ|)$