

# Brute Force Pattern Matching

We assume that the text is an array  $T[1 .. n]$  of length  $n$  and that the pattern is an array  $P[1 .. m]$  of length  $m \leq n$ .

We say that pattern  $P$  occurs with shift  $s$  in text  $T$ .

example:

text: abcabaabcbac and pattern: abaa,  $s = 3$

Except for the naive brute-force algorithm, each string-matching algorithm performs some preprocessing based on the pattern and then finds all valid shifts.

## Suffix and Prefix

Prefix and suffix are special cases of substring. A prefix of a string  $S$  is a substring of  $S$  that occurs at the beginning of  $S$ . A suffix of a string  $S$  is a substring that occurs at the end of  $S$ . e.g. "ab" is prefix of "abcca" and "cca" is suffix of "abcca".

## Algorithm

The naive algorithm finds all valid shifts using a loop that checks the condition  $P[1 .. m] = T[s + 1 .. s + m]$  for each of the  $n - m + 1$  possible values of  $s$ .

## Time and Space Complexity

Time complexity:  $O((n - m + 1) \times m) \approx O(n \times m)$

Space Complexity:  $O(1)$