

# Lab 11

## Problem 1

You are given a short story and a pattern. Your task is to find all occurrences of the pattern in the story using the brute-force string matching algorithm. The story is a string of text, and you need to find the starting indices where the pattern appears in the story.

### Input:

- **Story:** "Alice walked into the forest. Alice wondered what she would find. Suddenly, Alice saw a rabbit."
- **Pattern:** "Alice"

### Task:

Describe **step by step** how you would apply the brute-force string matching algorithm to find all occurrences of the pattern "Alice" in the given story.

---

## Problem 2

You are analyzing customer reviews to check if any of them mention a specific complaint keyword.

- The **pattern** you're searching for is: LATE
- The **text** from a feedback entry is:  
"THE DELIVERY WAS VERY LATE AND DAMAGED THE PACKAGE"

To efficiently find the pattern, you decide to apply **Horspool's Algorithm**, which is optimized for fast string searching using a **shift table** and right-to-left comparisons.

### Tasks:

1. **Build the shift table** for the pattern LATE using Horspool's rule:
  2. **Step by step**, describe how Horspool's Algorithm compares characters and performs shifts on the given text.
-

### Problem 3

You are implementing an auto-complete system that uses a **Trie** data structure. The system must support inserting words, checking if a word exists, and verifying whether any word starts with a given prefix.

Given the following list of words:

```
["cat", "cap", "can", "camp", "camera"]
```

#### Tasks:

1. Draw the **Trie structure** that results from inserting all the given words.
  2. Use the Trie to determine whether the prefix "cam" exists in the structure.
  3. Check whether the word "capo" exists in the Trie.
- 

### Problem 4

In a genetics research lab, scientists are analyzing DNA sequences to detect specific **harmful gene patterns** associated with increased risk of hereditary disorders. You are given a DNA sequence and a known **harmful gene pattern**, and your task is to find all locations where this pattern occurs in the sequence using an **efficient string matching algorithm**.

Each occurrence of the harmful pattern corresponds to a potential mutation site. If the pattern appears **more than twice** in the sequence, the DNA is considered at risk for **genetic instability**, which may contribute to cancer or other genetic disorders.

#### Input:

- **DNA sequence:**  
"ATCGTACGATCGGATCATCG"
- **Harmful gene pattern:**  
"ATCG"

#### Task:

1. Use an **efficient string matching algorithm** to find all starting indices (0-based) where the pattern occurs in the DNA sequence.
2. Describe **step by step** how your algorithm searches for the pattern in the DNA string. Based on the number of matches found, determine whether the sequence should be flagged for **genetic instability**.