

Introduction

The lesson pitches a brief overview of why and what you'll be learning next.

WE'LL COVER THE FOLLOWING



- `std::search` in C++14
- `std::search` in C++17
- What you'll learn in this Chapter?

`std::search` in C++14

`std::search` in C++14 offers a generic way to search for a pattern in a given range. The algorithm can be used not only for character containers but also for containers with custom types. This technique was, unfortunately, a bit limited as the performance was usually slow - it uses the naive matching algorithm, with the complexity of the size of the pattern times the size of the text.

`std::search` in C++17

With C++17 we get new `std::search` overloads that expose new and powerful algorithms like Boyer Moore variations that have linear complexity in the average case.

What you'll learn in this Chapter?

- How we can beat a naive search algorithm with pattern preprocessing.
- How you can use `std::search` to efficiently search for a pattern in a range.
- How to use `std::search` for custom types.

Let's get an overview of string matching algorithms for a start.

