

CS 61

<https://cs61.seas.harvard.edu/site/2021/>

Section

section 1: C++ data structures

This section covers important parts of the **C++ standard library**, especially container data structures. We also continue to explore **memory and data representation**.

C++'s Standard Template Library (STL)

C++ comes with a large library of useful data structures, including resizable arrays (`std::vector`), linked lists (`std::list`), ordered search trees (`std::map`), hash tables (`std::unordered_map`), and sets (`std::set` and `std::unordered_set`). It also comes with a library of useful algorithms, including `std::sort` (sorting) and `std::lower_bound` (binary searching). You may notice these structures and algorithms in handout code, and you may want to use these data structures yourself.

`std::vector`(resizable array)

`std::vector` represents an array of objects of type `T` that can **dynamically change size**. Access into `std::vector` is **as fast as access into a normal array**, even though elements can be added and removed at runtime.

[test](#)