ECS769P: Advanced OOP
Lab 6: STL Sequence Containers

Use the Standard Library sequence containers to solve these problems.

Exercise 1: Vector Capacity (basic)

Create a very large `std::vector` by iteratively adding elements. Periodically display the size and capacity as you add elements. How does the capacity increase?

Exercise 2: Fibonacci (core)

Write a program that stores the start (i.e. the first n elements) of the Fibonacci series in a container. When the user enters a integer i it should display the ith element of the series, either retrieving directly from the container OR if $i > n$ extending the series container as necessary.

Exercise 3: Deque vs List (core)

Write a program that demonstrates the performance differences between the standard library's `std::deque` and `std::list` containers in terms of: 1) insertion of elements at the ends; 2) insertion of elements in the middle; 3) random-access of elements. You will need to measure time elapsed, e.g. using clock.

Exercise 4: Tree Traversal (challenge)

Create a tree data structure by defining a class TreeNode. Each TreeNode stores an integer value `and` a vector of 0 or more child TreeNodes.

Write a function that displays all the integer values in a tree in *breadth-first* order, i.e. the root, then all nodes at depth 1, then all nodes at depth 2, etc. It should use a `std::queue` to store *open nodes* (the nodes it has yet to visit).

Write another function that does the same in *depth-first* order. What is a good choice of container for storing the open nodes?