1. Vector / Dynamic Array

- with allocator
- **Iterators:**
  - Iterator
  - Const Iterator
  - Reverse Iterator

2. Singly Linked List

- **Structure:**
  - Linked Nodes (with next pointer)
- **Iterators:**
  - Iterator
  - Const Iterator

3. Doubly Linked List

- **Structure:**
  - Linked Nodes (with both previous and next pointers)
- **Iterators:**
  - Iterator
  - Const Iterator

4. Stack

- **Structure:**
  - 4.1 Linked Implementation
  - 4.2 Array Implementation
  - 4.3 Template Container Implementation

5. Queue

- **Structure:**
  - 5.1 Linked Implementation
  - 5.2 Array Implementation
  - 5.3 Template Container Implementation

6. Deque

- **Structure:**
  - 6.1 Linked Implementation
  - 6.2 Array Implementation
- **Iterators:**
  - Iterator
  - Const Iterator

7. Set/Map

- **Structure:**
  - Binary Search Tree
- **Iterators:**
  - Const Iterator
- **Additional Features:**
  - Custom Comparator

8. Priority Queue

- **Structure:**
  - Binary Heap

9. Unordered Map/Set

- **Structure:**
  - Separate Chaining
- **Iterators:**
  - Const Iterator
- **Additional Features:**
  - Template Hasher

10. Unordered Map/Set (Preserves Insertion Order)

- **Structure:**
  - Separate Chaining
- **Iterators:**
  - Const Iterator
- **Additional Features:**
  - Template Hasher

11. Unordered Map/Set (Linear Probing)

- **Structure:**
  - Linear Probing
- **Iterators:**
  - Const Iterator
- **Additional Features:**
  - Template Hasher

12. Disjoint Set

- **Structure:**
  - Tree with parent array. Path compression.