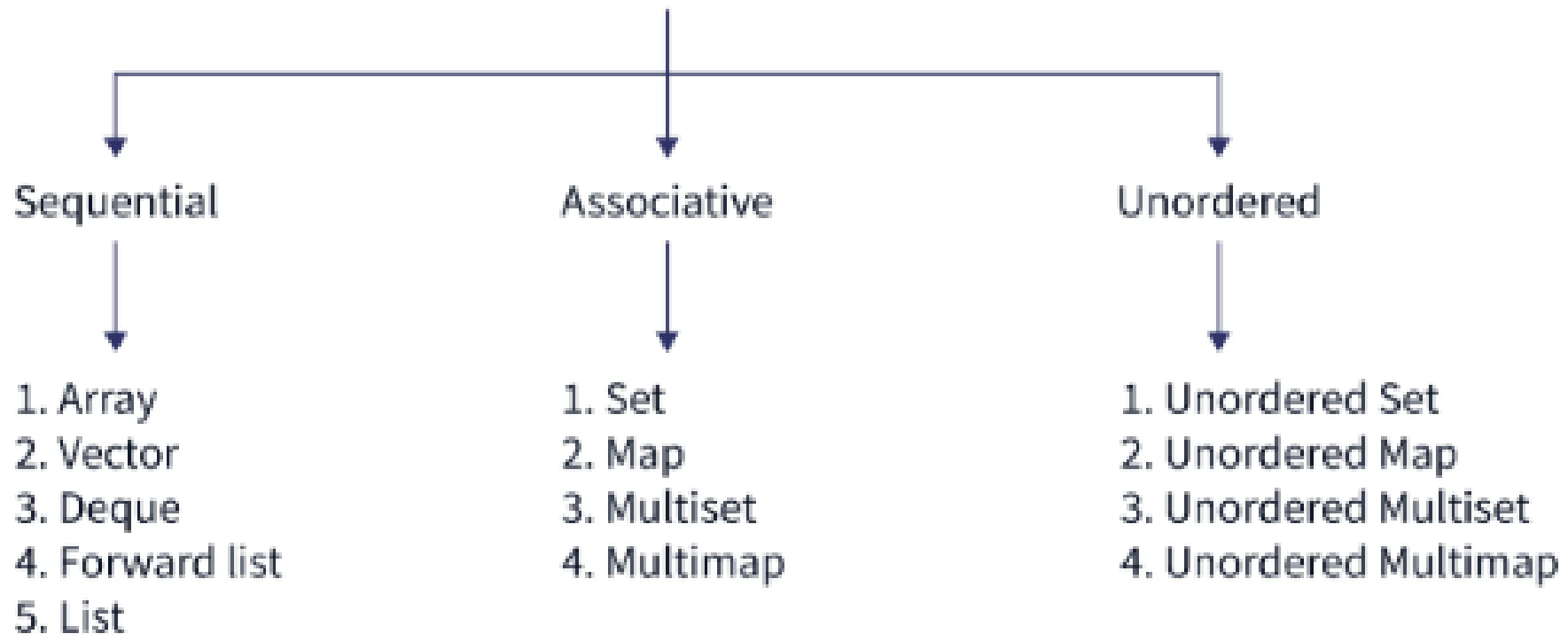




Containers en C++
<https://www.scaler.com/topics/cpp/containers-in-cpp/>

Types of Containers



Contenedores Secuenciales

`array<T, n>`



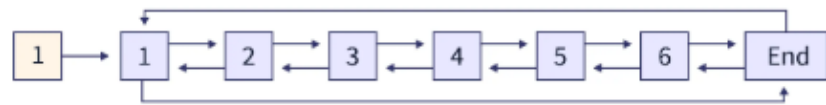
`vector<T>`



`deque<T>`



`list<T>`



`forward_list<T>`



SCALER
Topics

- Un Contenedor Secuencial en C++ es una colección ordenada del mismo tipo de datos en la que cada elemento se almacena en una posición específica. La posición de un elemento depende del lugar y momento de inserción de dicho elemento.

Contenedores Secuenciales

(a) Vectors (`std::vector`)

- Why use `vector` over arrays? Dynamic resizing, better memory management.
- Basic Operations: `push_back()`, `pop_back()`, `size()`, `at()`, `begin()`, `end()`

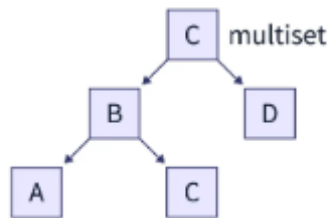
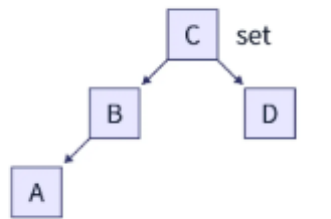
(b) Lists (`std::list`)

- Doubly linked list, efficient insertions/deletions.
- Operations: `push_front()`, `push_back()`, `pop_front()`, `pop_back()`

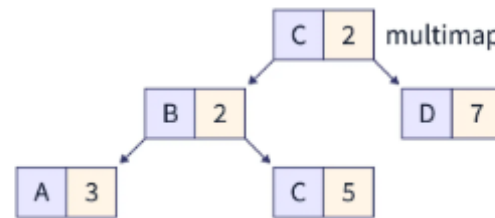
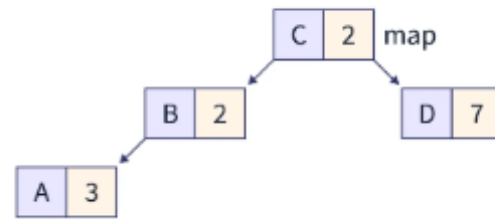
(c) Deques (`std::deque`)

- Double-ended queue, efficient insertions/removals from both ends.
- Example: Similar to `vector`, but allows `push_front()` efficiently.

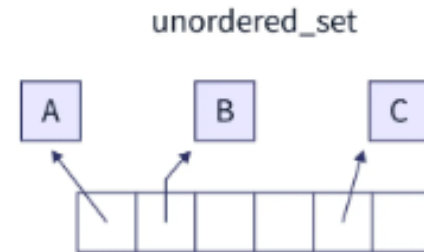
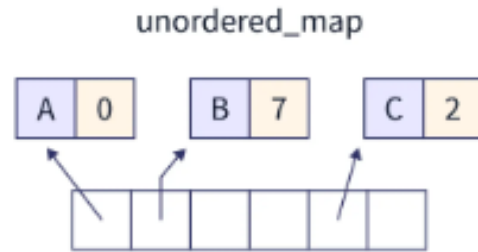
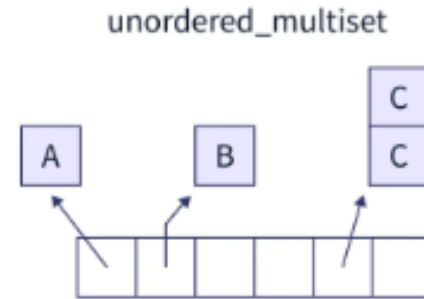
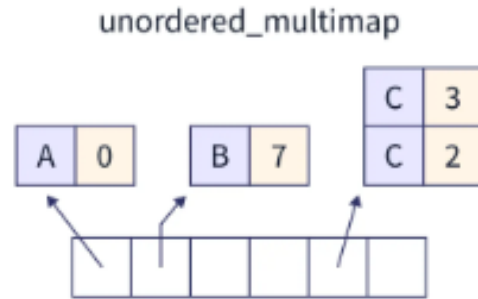
Contenedores Asociativos



SCALER
Topics



- Los contenedores asociativos son aquellos en los que la ubicación de un elemento depende del valor del elemento. El orden de inserción de los elementos no se tiene en cuenta para determinar la posición del elemento. A los elementos de un contenedor asociativo se accede mediante claves, también llamadas claves de búsqueda.



SCALER
Topics

Contenedores Desordenados (asociativos)

- Los contenedores desordenados o contenedores asociativos desordenados son aquellos contenedores en los que la posición de los elementos no importa. Los elementos de los contenedores desordenados no se almacenan según el orden de su inserción o sus valores. Si se almacenan n elementos en un contenedor desordenado, el orden de sus posiciones será indefinido, e incluso podría cambiar con el tiempo. Se puede acceder a los elementos de estos contenedores mediante hash.

Adaptadores de Contenedores

stack

Último en entrar primero en salir (UEPS).

queue

Primero en entrar, primero en salir (PEPS).

priority_queue

El elemento de mayor prioridad siempre es el primero en salir.

<array>

<vector>

<list>

<deque>

<queue>

Contiene tanto a queue como a priority_queue.

<stack>

<map>

Contiene tanto a map como a multimap.

<set>

Contiene tanto a set como a multiset

Adaptadores de Contenedores

5. Best Practices & Use Cases

- Choosing the right container:
 - Fast lookups? `unordered_map`, `set`
 - Ordered elements? `vector`, `map`
 - Frequent insertions/deletions? `list`, `deque`
 - Need a stack/queue behavior? `stack`, `queue`

Preguntas?

