Containers	Functions	Iterators

Array->	.fill(0)
	.at( ) or a[]
	.front() or arr.at(0)
	.back() or arr.at(arr.size()-1)
	.empty()
	.sizeof()
	.size() //size of the array
		.begin() //point to the first element
		
		.rbegin() //point to the last element
		
		.end() //point to next element after last element
		
		.rend() //point to next element before first element

Vectors →	.push_front() .push_back() .pop_front() .pop_back() .fill(0) .at( ) or a[] .front() or arr.at(0) .back() or arr.at(arr.size()-1) .empty() .erase(begin,end) .clear() .size()	Same as Array
Deque →	Rest Same as Vector	Same as Array
List →	.remove(2) Rest Same as Vector	Same as Array
Set → -unique acceding order for set -unique random order for unordered_set -repeating values in ascending order with multi-set	.insert() .find() .count() .empty() .erase(begin,end) .clear() .size()	Same as Array
Map → -same as set with key and value combo -map,unordered_map,multimaps same as set	.first .second .at(key) or a[key] Rest Same as Set	Same as Array
Stack → -LIFO (Last in First Out)	.push() or emplace .pop() .top() .empty() .size()	XX_NO ITERATORS_XX

Queue →
-FIFO Operation ()
	.push()
	.pop()
	.front() // first element
	.back() //last element
	.empty()
	.size()
		XX_NO ITERATORS_XX

Priority Queue → -Stores all in Sorted order and dose it in log N -Max & Min Priority Queue	Same as Stack	

.find() → This return the value itself that we are searching if it is present in the set, if it is not present in the set it returns the location in integer form like 7^{th} position and this position is the .end() position.

.end() → This also returns the position in integer format like find()