

Oversikt over Containere:

	Sekvensiell	<array>	Samme som <code>vector</code> , bare med <i>fast/konstant lengde</i>
	Assosiativ	<bitset>	Bool-array, der <i>en bool tar ett bit</i>
	Sekvensiell	<deque>	Samme som <code>vector</code> , bare at <i>også kan legge til/ta vekk elementer forrest</i>
	Sekvensiell	<forward_list>	Samme som <code>list</code> , bare at <i>kun pekere til den neste (og ikke til den forrige)</i>
Vi bruke	Sekvensiell	<list>	Dobbelt-linked/toveis liste, kan ta ut/sette inn hvor som helst i listen
Vi bruke	Assosiativ	<map>	Unike nøkkel og tilhørende data sortert (logaritmisk)
Vi bruke	Adapter	<queue>	Setter inn elementer bakerst, tar kun ut forrest
	Assosiativ	<set>	<i>Unike verdier (uten tilhørende data som i <code>map</code>) lagres sortert</i> (logaritmisk)
			(ny i C++20)
Vi bruke	Adapter	<stack>	Setter kun inn og tar ut elementer forrest
	Assosiativ	<unordered_map>	<i>Unike nøkkelpar delvis usortert</i> (konstant)
	Assosiativ	<unordered_set>	<i>Unike verdier lagres delvis usortert</i> (konstant)
Vi bruke	Sekvensiell	<vector>	Dynamisk array, kun legge til/ta vekk elementer bakerst
	Assosiativ	<multimap>	Samme som <code>map</code> , bare at trenger <i>ikke</i> å være <i>unike</i> (logaritmisk)
	Assosiativ	<multiset>	Samme som <code>set</code> , bare at trenger <i>ikke</i> å være <i>unike</i> (logaritmisk)
	Adapter	<priority_queue>	Setter inn på en slik måte at <i>det største kan tas ut igjen først</i>
	Assosiativ	<unordered_multimap>	Samme som <code>unordered_map</code> , bare at trenger <i>ikke</i> å være <i>unike</i> (konstant)
	Assosiativ	<unordered_multiset>	Samme som <code>unordered_set</code> , bare at trenger <i>ikke</i> å være <i>unike</i> (konstant)