Container der STL

C++ Übung am 16. Juni 2016

Container

`Container`	`Beschreibung`	`Einfügen/ Löschen`
`Vector`	Eindimensionaler Vector aus der Mathem.	Am Ende effizient
`Deque`	Eindimensionaler Vector aus der Mathem.	Am Ende und am Anfang effizient
`List`	Eindimensionaler Vector aus der Mathem.	Überall effizient

Container

Container	Beschreibung	Identische Objekte
`Set`	Geordnete Menge von Objekten	Nein
`Map`	Geordnete Menge von Tupeln von Objekten	Nein
`Multiset`	Menge von Objekten	Ja
`Multimap`	Menge von Tupeln von Objekten	Ja
`Unordered set`	Ungeordnete Menge von Objekten	Nein
`Unordered map`	Ungeordnete Menge von Tupeln von Objekten (Hash)	Nein

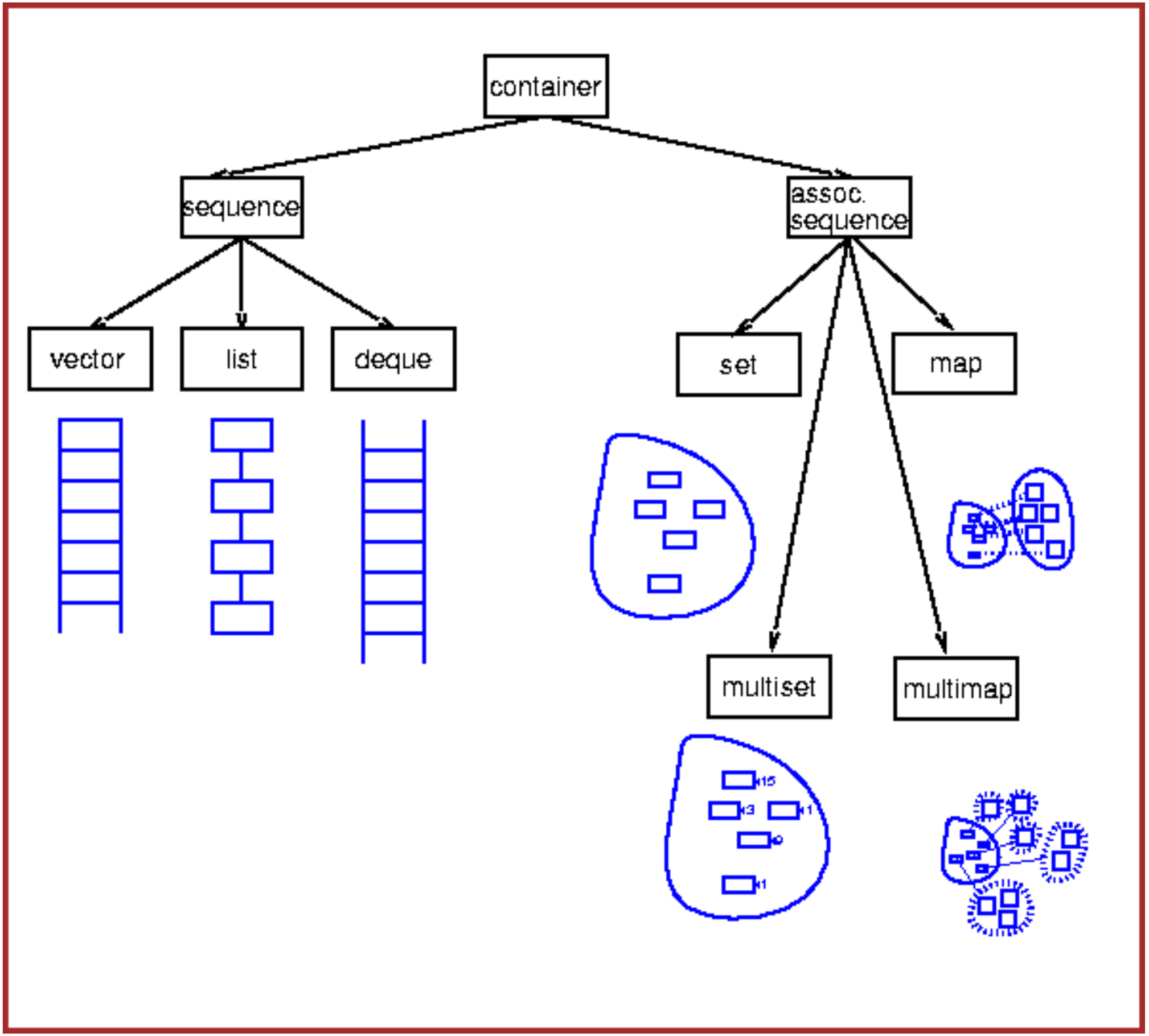
Laufzeitverhalten: Effizienz/ Performanz/ O-Notation

	Komplexität	**Bezeichnung**	**Bedeutung**
1	O(1)	konstant	unabhängig von Zahl der Elemente
2	O(log n)	logarithmisch	bei linearer Erhöhung der Elemente immer langsamerer Anstieg
3	O(n)	linear	Zeit steigt proportional zur Elementzahl
4	O(n*log n)	n*log n	Zeit steigt loglinear zur Elementzahl
5	$O(n^2)$	quadratisch	doppelte Anzahl, vierfache Zeit
6	$O(2^n)$	exponentiell	super schlecht!

Containervergleich

Aufgabe	<array>	<vector>	<list>	<deque>
C++11	x			
Zugriff []	O(1)	O(1)	**kein wahl-freier Zugriff**	O(n)
einfügen/ löschen	-	am Ende: O(1) sonst: O(n)	sau schnell überall: O(1)	vorn/ hint: O(1) sonst: O(n)
suchen	find (algorithm) O(n)	find (algorithm) O(n)	find (algorithm) O(n)	find (algorithm) O(n)
sortieren	sort (algorithm) O(n*log n)	sort (algorithm) O(n*log n)	**eigenes Sortieren** O(n*log n)	sort (algorithm) O(n*log n)

Aufgabe	<set>	<map>	<unordered_set>	<unordered_map>	<multiset>	<multimap>
C++11			x	x		
Zugriff []	nein	über key O(log n)	nein	über key O(1) - O(n)	nein	nein
einfügen/ löschen	schnell, O(log n)	schnell, O(log n)	sehr schnell, O(1)	sehr schnell, O(1)	schnell, O(log n)	schnell, O(log n)
suchen	**eigenes Suchen** O(log n)	**eigenes Suchen** O(log n)	eigenes Suchen O(1)	eigenes Suchen O(1)	**eigenes Suchen** O(log n)	**eigenes Suchen** O(log n)
sortieren	schon sortiert	schon sortiert	nicht sortierbar: Hash	nicht sortierbar: Hash	schon sortiert	schon sortiert



Grafik ausgeliehen von Roman Capsamun und Nicola Greth

Klausurübung

Schreiben Sie ein C++ Programm, das die eigene Klasse Words definiert. Die Idee ist, dass diese Klasse in einem wstring Vector alle Wörter eines Textes speichert. In Ihrem private Bereich soll ein Vektor definiert werden, der beliebige wstrings speichern kann und eine Methode is_element, die testet ob ein Wort bereits im Vektor eingetragen ist. Die Klasse Words definiert eine public Methode insert(wstring word), die ein Element in das interne Memory der lexikon Klasse hinzufügt. Die Klasse Words definiert eine public Methode print(), die ein Element des internen Memories der Words Klasse ausdruckt.