

метод	dsa	dynamic_array	ForwardList	DoubleLinkedList	TemplateQueue	TemplateStack	BinarySearchTree	-	HashMap	-	Trie
	std	vector	forward_list	list	queue	stack	-	set / map	unordered_map	unordered_set	
push_back		O(1) аморт.	O(1)	O(1)	O(1)	O(1) / O(1) аморт.	-	-	-	-	с m е означена дължината на думата
pop_back		O(1)	-	O(1)	-	O(1)	-	-	-	-	
push_front		-	O(1)	O(1)	-	-	-	-	-	-	
pop_front		-	O(1)	O(1)	O(1)	-	-	-	-	-	
insert		O(n)	O(1)	O(1)	-	-	O(log(n)) [O(n)]	O(log(n))	O(1) аморт. [O(n)]	O(m)	
erase		O(n)	O(1)	O(1)	-	-	O(log(n)) [O(n)]	O(log(n))	O(1) [O(n)]	O(m)	
at		O(1)	-	-	-	-	-	-	O(1) [O(n)]	-	
front		O(1)	O(1)	O(1)	O(1)	-	-	-	-	-	
back		O(1)	O(1)	O(1)	-	O(1)	-	-	-	-	
begin		O(1)	O(1)	O(1)	-	-	-	O(1)	O(1)	-	
end		O(1)	O(1)	O(1)	-	-	-	O(1)	O(1)	-	
find		O(n)	O(n)	O(n)	-	-	O(log(n)) [O(n)]	O(log(n))	O(1) [O(n)]	O(m)	
size		O(1)	O(1)	O(1)	O(1)	O(1)	-	-	O(1)	-	
Легенда:											
O(1)	Сложностите са във вида									Забележка: Итерирането през unordered_map и unordered_set става за време O(k+n), където k е броят на кофите използвани за реализиране на структурата.	
O(1) амортизир.	Сложност в най-лошия случай										
O(log(n))				или							
O(sqrt(n))	Сложност в средния случай [Сложност в най-лошия случай]										
O(n)				или							
O(n^2)	Сложност в най-лошия случай от std / Сложност в най-лошия случай от dsa										
не съществува в std											