

Gyakran használt STL tárolók

Osztály neve	#include	Megnevezés	Leírás
<code>vector</code>	<code><vector></code>	vektor	Elemek tárolása közvetlen indexelt elérésre.
<code>list</code>	<code><list></code>	lista	Láncolt, kétirányú lista.
<code>map</code>	<code><map></code>	asszociatív	Kulcs-elem párok ábrázolása: kulcs \mapsto elem.
<code>set</code>	<code><set></code>	halmaz	Minden elem egyszer szerepelhet benne.
<code>multiset</code>	<code><set></code>	zsák	Minden elem többször is szerepelhet benne (ez az elem multiplicitása).
<code>deque</code>	<code><deque></code>	*	Kétvégű sor; konstans időben: elem elérése, elejére/végére beszúrás.
<code>stack</code>	<code><stack></code>	verem	LIFO adatszerkezet adapter.
<code>queue</code>	<code><queue></code>	sor	FIFO adatszerkezet adapter.
<code>priority_queue</code>	<code><queue></code>	prioritásos sor	Összehasonlítható elemeket tárol, a legnagyobb vehető ki először.

Gyakran használt STL algoritmusok

	Egyszerűsített szignatúra		Leírás
Fun1	<code>for_each</code>	<code>(In, In, Fun1 f);</code>	Minden elemre végrehajtja az <code>f</code> függvényt.
InIt	<code>find</code>	<code>(In, In, const T&);</code>	Megkeresi az elemet. Ha nincsen, <code>end()</code> az eredmény.
InIt	<code>find_if</code>	<code>(In, In, Pred1 pred);</code>	Megkeresi az első elemet, amelyre <code>pred</code> teljesül.
int	<code>count</code>	<code>(In, In, const T&);</code>	Megszámolja, hányszor fordul elő az érték.
int	<code>count_if</code>	<code>(In, In, Pred1 pred);</code>	Megszámolja, hány elemre teljesül <code>pred</code> .
OutIt	<code>copy</code>	<code>(In, In, Out out);</code>	Átmásolja az elemeket az <code>out</code> területre.
OutIt	<code>transform</code>	<code>(In, In, Out out, Fun1 f);</code>	Elemenként alkalmazza az <code>f</code> függvényt, eredmény: <code>out</code> területen.
void	<code>replace</code>	<code>(Forw, Forw, T old, T new);</code>	Kicseréli az <code>old</code> értékeket <code>new</code> értékekre.
void	<code>fill</code>	<code>(Forw, Forw, const T&);</code>	Feltölti a tárolót a megadott elemmel.
void	<code>generate</code>	<code>(Forw, Forw, Fun0 gen);</code>	Feltölti a tárolót <code>gen</code> által előállított értékekkel.
FIt	<code>unique</code>	<code>(Forw, Forw);</code>	Csak egy marad egymás utáni egyező elemekből, többi: végére.
FIt	<code>unique</code>	<code>(Forw, Forw, Pred2 pred);</code>	Mint az előző; az egyezést itt <code>pred</code> írja le.
void	<code>reverse</code>	<code>(Bidi, Bidi);</code>	Megfordítja az intervallumot.
FIt	<code>partition</code>	<code>(In, In, Pred1 pred);</code>	Az elejére csoportosítja azokat, amelyekre <code>pred</code> igaz.
void	<code>sort</code>	<code>(Rand, Rand);</code>	Növekvő sorrendbe rendezi az intervallumot.
void	<code>sort</code>	<code>(Rand, Rand, Pred2 cmp);</code>	Mint az előző; az elemeket <code>cmp</code> hasonlítja össze.
void	<code>stable_sort</code>	<code>(Rand, Rand);</code>	Mint <code>sort</code> ; az azonos elemek sorrendje megmarad.
void	<code>stable_sort</code>	<code>(Rand, Rand, Pred2 cmp);</code>	Mint az előző; az elemeket <code>cmp</code> hasonlítja össze.
void	<code>merge</code>	<code>(InA, InA, InB, InB, Out);</code>	Rendezett tárolók összefésülése <code>operator<</code> reláció szerint.
void	<code>merge</code>	<code>(In..., Out, Pred2 cmp);</code>	Mint az előző; az elemeket <code>cmp</code> hasonlítja össze.
FIt	<code>min_element</code>	<code>(In, In, Pred2 cmp);</code>	A legkisebb elem iterátora.
FIt	<code>max_element</code>	<code>(In, In, Pred2 cmp);</code>	A legnagyobb elem iterátora.
FIt	<code>remove</code>	<code>(Forw, Forw, const T&)</code>	A "törölt" elemek a végére kerülnek. Vissza: új <code>end()</code> .

STL tárolók gyakori műveletei

Egyszerűsített szignatúra			Leírás	vec	list	deq	map	set	mset
It	begin	()	A tároló eleje.	X	X	X	X	X	X
It	end	()	A tároló vége.	X	X	X	X	X	X
RIt	rbegin	()	A megfordított tároló eleje.	X	X	X	X	X	X
RIt	rend	()	A megfordított tároló vége.	X	X	X	X	X	X
int	size	()	A tárolt elemek száma.	X	X	X	X	X	X
int	max_size	()	A tároló maximális mérete.	X					
size_t	capacity	()	A tároló kapacitása.	X					
size_t	reserve	(int m)	A kapacitás növelése m-re.	X					
T&	operator[]	(int n)	A tároló n-edik eleme.	X		X	X		
T	operator[]	(Key k)	A halmaz k kulcsú eleme.				X		
bool	empty	()	A tároló üres-e?	X	X	X	X	X	X
T	front	()	A tároló első eleme.	X	X	X			
T	back	()	A tároló utolsó eleme.	X	X	X			
void	push_front	(const T&)	Az elem a tároló elejére kerül.		X	X			
void	push_back	(const T&)	Az elem a tároló végére kerül.	X	X	X			
void	pop_front	()	A tároló első elemének törlése.		X	X			
void	pop_back	()	A tároló utolsó elemének törlése.	X	X	X			
void	swap	(Tároló&)	Tároló felcserélése hasonló tárolóval.	X	X	X	X	X	X
It	erase	(It)	A mutatott elem törlése.	X	X	X			
void	erase	(It)	A mutatott elem törlése.				X	X	X
It	erase	(It, It)	Az intervallum törlése.	X	X	X	X	X	X
void	clear	()	A tároló kiürítése.	X	X	X	X	X	X
It	insert(It pos, const T&)		Az elem beillesztése pos után.	X	X	X			
It	insert	(Key, const T&)	Új kulcs/elem beillesztése.				X		
It	insert	(const T&)	Az elem beillesztése.					X	X

STL függvényobjektumok

Név	Jelentés	Név	Jelentés
plus	$e_1, e_2 \mapsto e_1 + e_2$	equal_to	$e_1, e_2 \mapsto e_1 = e_2$
minus	$e_1, e_2 \mapsto e_1 - e_2$	not_equal_to	$e_1, e_2 \mapsto e_1 \neq e_2$
multiplies	$e_1, e_2 \mapsto e_1 \times e_2$	less	$e_1, e_2 \mapsto e_1 < e_2$
divides	$e_1, e_2 \mapsto e_1 / e_2$	greater	$e_1, e_2 \mapsto e_1 > e_2$
modulus	$e_1, e_2 \mapsto e_1 \bmod e_2$	less_equal	$e_1, e_2 \mapsto e_1 \leq e_2$
negate	$e \mapsto -e$	greater_equal	$e_1, e_2 \mapsto e_1 \geq e_2$
binder1st(v)	$(e_1, e_2 \mapsto f(e_1, e_2)) \mapsto (e_2 \mapsto f(v, e_2))$	binder2nd(v)	$(e_1, e_2 \mapsto f(e_1, e_2)) \mapsto (e_1 \mapsto f(e_1, v))$
ptr_fun(f)	f		

Jelmagyarázat:

- In, Forw, Out, Bidir, Rand: Input/Forward/Output/Bidirectional/RandomIterator
- FunN: N argumentumú függvény, visszatérési értéke a tároló egy eleme
- PredN: predikátum, N argumentumú függvény, visszatérési értéke logikai érték
- T: a tárolt érték típusa
- Key: map tárolónál a kulcs típusa
- Tároló: az adott tárolóval megegyező típus
- size_t: tárolási méret típusa, általában typedef egy unsigned típusra

Megjegyzés: az asszociatív tárolók konstans iterátorokat adnak vissza.