

**Pilha: fila com disciplina LIFO***Complexidades em todos os casos*

Operação	Pilha em Vector	Pilha em Lista Ligada
<b>isEmpty</b>	$O(1)$	$O(1)$
<b>size</b>	$O(1)$	$O(1)$
<b>top</b>	$O(1)$	$O(1)$
<b>push</b>	$O(1)$	$O(1)$
<b>pop</b>	$O(1)$	$O(1)$

**Fila: fila com disciplina FIFO***Complexidades em todos os casos*

Operação	Fila em Vector	Fila em Lista Ligada
<b>isEmpty</b>	$O(1)$	$O(1)$
<b>size</b>	$O(1)$	$O(1)$
<b>enqueue</b>	$O(1)$	$O(1)$
<b>dequeue</b>	$O(1)$	$O(1)$

**Lista: acesso por posição***Complexidades da lista ligada com  $n$  elementos*

Operação	Melhor C.	Pior C.	C. Esperado
<b>isEmpty, size</b>	$O(1)$	$O(1)$	$O(1)$
<b>getFirst, getLast</b>	$O(1)$	$O(1)$	$O(1)$
<b>get</b>	$O(1)$	$O(n)$	$O(n)$
<b>addFirst, addLast</b>	$O(1)$	$O(1)$	$O(1)$
<b>add</b>	$O(1)$	$O(n)$	$O(n)$
<b>removeFirst, removeLast</b>	$O(1)$	$O(1)$	$O(1)$
<b>remove</b> (por posição)	$O(1)$	$O(n)$	$O(n)$
<b>find, remove</b> (por elemento)	$O(1)$	$O(n)$	$O(n)$
<b>iterator</b>	$O(1)$	$O(1)$	$O(1)$

**Iterador bidireccional**

Operação	Em Todos os Casos
<b>hasNext</b>	$O(1)$
<b>next</b>	$O(1)$
<b>rewind</b>	$O(1)$
<b>hasPrevious</b>	$O(1)$
<b>previous</b>	$O(1)$
<b>fullForward</b>	$O(1)$

**Fila com prioridade: acesso por prioridade***Complexidades em todos os casos*

Operação	Fila em Vector	Fila em Lista Ligada
<b>isEmpty</b>	$O(1)$	$O(1)$
<b>size</b>	$O(1)$	$O(1)$
<b>enqueue</b>	$O(1)$	$O(1)$
<b>dequeue</b>	$O(1)$	$O(1)$

**Dicionário Ordenado: acesso por chave***Complexidades quando o dicionário ordenado tem  $n$  entradas*

	Melhor Caso	Pior Caso	Caso Esperado
Pesquisa	$O(1)$	$O(n)$	$O(\log n)$
Inserção	$O(1)$	$O(n)$	$O(\log n)$
Remoção	$O(1)$	$O(n)$	$O(\log n)$
Mínimo	$O(1)$	$O(n)$	$O(\log n)$
Máximo	$O(1)$	$O(n)$	$O(\log n)$
Obter Iterador	$O(1)$	$O(n)$	$O(\log n)$
Percurso Ordenado	$O(n)$	$O(n)$	$O(n)$

**Pesquisa**      **Inserção**      **Remoção**      **Mínimo**      **Máximo**

	Pior Caso	Caso Esperado
árvore sem restrições	$n$	$\log n$
AVL	(1.44) $\log n$	$\log n$
vermelha e preta	(2) $\log n$	$\log n$

**Percurso**      **Percurso Ordenado**

	"qualquer caso"
"qualquer" árvore	$n$