Comparing *Linear* Container Data Structures

(all: `template <typename ItemType>`)
## Major *Linear* Container Classes

<table>
<thead>
<tr>
<th></th>
<th>Properties</th>
<th>Applications</th>
<th>Implementations</th>
</tr>
</thead>
</table>
| **List** | add/delete/retrieve anywhere        | • General ordered or unordered collections of objects (windows in a GUI; names in a roster; objects in a drawing program)  
• Usually we want to access some or all of the elements many times *without* removing them. | • Linked list data structure  
• Array is almost always a poor choice |
| **Stack** | **LIFO:** add/delete/retrieve at one end (the “top”) | • We remove objects as we access them  
• Activation records; backtracking algorithms (or anything for which recursion might be appropriate) | • Array (if max size is known at time of instantiation)  
• Linked list data structure |
| **Queue** | **FIFO:** add at back; remove/retrieve from front | • We remove objects as we access them  
• Managing access to shared services or objects; used to perform certain types of traversals of advanced data structures  
• Simulations of real shared access situations (e.g., to gather statistics on service times and/or look for bottlenecks) | • Array (if max size is known at time of instantiation)  
• Linked list data structure |