Queue is first in, first out (FIFO)

Stack is last in, first out (LIFO)

<https://www.quora.com/Under-what-circumstances-would-a-web-developer-need-to-use-data-structures-such-as-Linked-Lists-BST-and-Graphs>

***(Doubly) Linked List***  
**Properties**  
Linked Lists store sequential data and the grow and shrink in linear time. They gain this capability at the expense of random access (you cannot view values in the middle of the linked list without transverse part of it).  
  
**Use Case Example**  
One common use of a Linked List would be to implement a Queue. A Queue takes values and allows them to be removed in the order they were added. A Linked List based queue can continually grow without ever needing to resize. Imagine a website that took requests from customers and handled them in the order they were made. You could store this in a Linked List based queue and feel safe in the knowledge that no matter how many requests your customers made adding and removing items from the queue would always take linear time (unlike an array based implementation which would occasionally need to use linear time to resize).

***Graph***  
**Properties**  
A graph is a data structure that models a network of objects. That is to say some set of objects each of which has some sort of connection to zero or more of the same type of object. This is an incredibly common structure for real world data to have.  
  
**Use Case Example**  
Your building a social network like Facebook. Your users can have friends. You notice that your data constitutes a graph. An account is a vertex (aka node) and a friendship is an edge (aka link). To answer questions about relationships between your users you need to run graph algorithms. Want to know if Bob is a friend of a friend of Jim? Run Breadth First Search with a max depth of 2 and see if you can find  path between them. You can also use data gleaned form this graph to help suggest friends, gleen insights for advertisers, ect.

***Binary Searh Tree***  
**Properties**  
Binary search trees store order-able data in such a way that it can be retrieved in logarithmic time. You can also transverse a BST in sorted order.  
  
**Use Case Example**  
You need to store information about all the users who are currently online in your chat app. You want to be able to print out there names in sorted order and search through there names to find if a specific user is online. A binary search tree allows you to do both of these operations quickly.