This week we observed a data structure known as a linked list. A linked list is a type of list that contains data-held items known as “pointers” that point to other data containing items.

In every programming language, there is some sort of array class that can hold many objects. These arrays (also known as lists or vectors in some languages) are great for storing small amounts of data. There is a problem, however, when adding or subtracting from these arrays. This is due to how the data is stored in memory. When adding or subtracting and element, each other element needs to be shifted in its memory location to ensure contingency in the array. This can be a very time-consuming process and as such does not work well for large lists.

This is where a linked list becomes useful. This type of list doesn’t rely on each element’s memory location to be in order in the collection. Each element also has a pointer that just points to the next memory location of whatever object should be next. This cuts down the time needed when adding or subtracting elements from the collection.

A singly linked list contains only a pointer for whatever element should come next. A doubly linked list contains pointers for both the next element and the previous element. Because of this, a singly linked list can only be traversed in one, forward direction if a developer needed to iterate over it, whereas the doubly linked list could be traversed in both directions.

When choosing which type of list to use, because of the number of pointers, a singly linked list will be smaller, but a doubly linked list will have better performance when searching. Choose the former if memory will be an issue on your platform, otherwise, go with the latter.

Linked lists are best used when dealing with very large data sets. Something like an e-commerce website would probably use a linked list. Arrays would be much too slow when adding or removing a product!

When it comes to searching a linked list, instructions are needed so the program knows what to do. This week we used C++’s compare() method. This method can be used when searching many different types of data sets. Not necessarily linked lists exclusively. This was just the safest method to use in our instance.