

Supplementary Sheet #1

Arrays, Array Lists and Linked Lists

The three data structures that Problem Set #1 covers are: arrays, array lists and linked lists.

	Arrays	Array Lists	Linked Lists
Size	Fixed	Variable	Variable
Contiguity	Contiguous	Contiguous	Non-Contiguous
Access Time	Random Access	Random Access	Serial Access

It is very important that you understand the differences between arrays, array lists and linked lists. The first difference is that of capacity. Arrays can hold only a fixed capacity, which must be predefined. Array lists and linked lists can hold as many items as required. The difference between array lists and linked lists is that, like arrays, array lists store their data contiguously in memory.

What does contiguously mean? It is a very important term, that means the array is stored together in a single block of memory. So if the location of the first item is the memory location 52, the location of the second item is, therefore, memory location 53. The location of the third item is memory location 54, and so on. What does this mean? It means that the computer can work out the memory location of any index you give it, by adding the index to the start memory location. This means to find the first item, where the start of the array is stored in memory location 52, the computer will do $(52 + 0 = 52)$ and look in memory location 52. To find the fiftieth item, it will do $(52 + 49 = 101)$. This means the computer can just as quickly get to the first item as the second item. That is why arrays and array lists are called random-access data structures. You can pick any random index, and the computer will be able to find it just as fast. Linked lists are not stored contiguously in memory, they are stored all over the place. Therefore, to get to the 5th item, the computer needs to find the 1st, 2nd, 3rd and 4th item. That is why linked lists are serial access data structures. Serial access data structures are slow when it comes to retrieving data stored at any random index. This is why unless you are going to accessing data stored serially, linked lists are a bad choice.

An array list tends to be implemented using arrays. Java's standard implementation of the array list starts with a default capacity of 10. Array lists will tend to increase their storage after a certain criterion is met. E.g. one implementation of an array list might double its storage capacity if it becomes full. Linked lists, meanwhile, are usually implemented internally using a node class. The node class stores the data item, and a reference to the next node. If the node is the last node, then its reference will be *null*. The linked list class will then store a reference to the first node.