Kristen Orue

Exploration: Arrays

Here in Canvas, please submit your answers to the following questions.

1. How is an array linear and ordered?

An array is linear because an array is a data structure containing values that are referenced by a numerical index starting at 0 and ending at whichever value one sets it to. As a concept it is both linear when drawn and ordered due to the indexes mentioned in the previous reasoning.

1. What are the six fundamental operations we can perform with arrays?

Store and retrieve which we get due to the nature of arrays automatically. We can also insert, delete, search, and sort.

1. How does a computer calculate the actual memory address for storing something in an array? (What is the formula?)

We can refer to a value in an array by it’s numeric index starting at 0 to whichever value you set.

1. Why do arrays typically hold values all of the same data type (size) ?

Because when we create an array, we calculate the memory address to store something in it and therefore we specific the element size.

1. What is the algorithmic complexity of array storage and retrieval, and why?

Because although arrays are labeled starting at 0 ending at whichever index a programmer assigns it to, there is a real dedicated set of memory that the computer stores it in that isn’t simply 0-5 when you create an array with 5 elements. The code np.ndarray(5, ‘I’) allocates five 32-bit integers/4 bytes integers. So since each box holds on integer/4 bytes. So every time a programmer writes a variable name for their array, it’s really just our human view of a computer memory address.

1. What is the algorithmic complexity of  inserting values into the middle of an array, or deleting a value from the middle of an array?

There is a mathematical equation that can help better understand this complexity by taking the base address adding the conceptual address ex:[2] and multiplying it by 4 (4bytes of dedicated storage per element) and that will be the actual address where the number is stored. Since there is a dedicated amount of memory (4 bytes per box and there are 5 boxes when a programmer goes to insert a number in the middle it must move one number out and it seems the last number in the array gets bumped to oblivion. Similarly if you were to delete the middle value, the last two values shift to the left. For the same reason: Ordered and linear.