Data Structures: Takeaways ↪

Syntax

- Inserting an element into a list at the given index:

```
players.insert(0, "Jeff Adrien")
```

- Retrieving an element in a 2D array:

```
arr[2][3]
```

- Retrieving the value of a key within a dictionary:

```
city_population["Boston"]
```

Concepts

- A data structure is a way of organizing data. Specifically data structures are concerned with organization of data within a program. Lists and dictionaries are some examples of data structures, but there are many more.

- An array is a list that can contain items, which occupy a specific slot. Arrays cannot expand beyond their initial size. For example, if we create an array of size 10, it can only hold 10 elements.

- When we delete or add an element to the array, each element has to be shifted, which can make those operations quite costly.

- Dynamic arrays are a type of array that we can expand to fit as many elements as we'd like. Dynamic arrays are much more useful in data science than fixed-size arrays.

- A list is a one-dimensional array because they only go in one direction. One-dimensional arrays only have a length and no other dimension.

- A two-dimensional array has a height and a width, and has two indexes.

- In data science, we call one-dimensional arrays vectors and two-dimensional arrays matrices.

- The time complexity of a two-dimensional array traversal is   where m is the height of our array, and n is the width.

- A hash table is a data structure that stores data based on key–value pairs. A dictionary is the most common form of a hash table.
- A hash table is a large array; however, a hash table is a clever construct that takes advantages of accessing elements by index and converts the keys to indexes using a hash function.
- A hash function accepts a key as input and converts it to an integer index.
- Accessing and storing data in hash tables is very quick; however, using a hash table uses a lot of memory.

Resources

- [Hash Table](#)