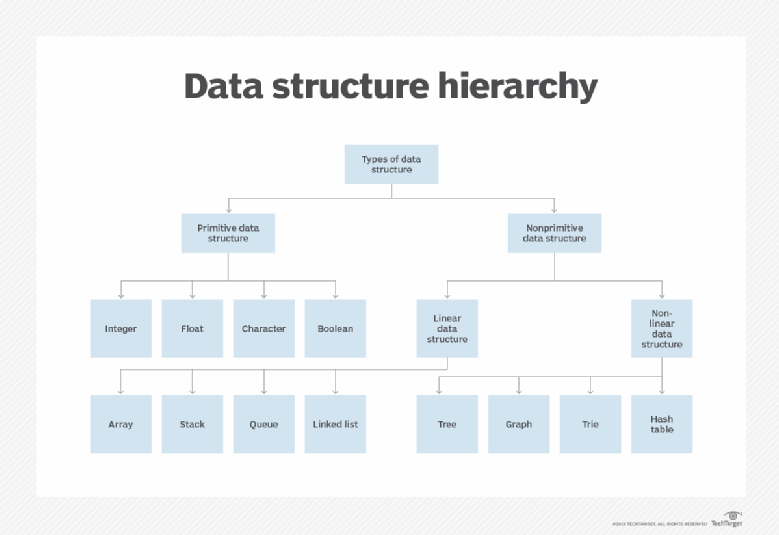
This is a entry level class, so it is important to tackle large ideas at a basic level. The hierarchy of data structure and brief discussions of each will be covered. This is a very dense topic with many different avenues to follow. This paper will not go into deep learning, but I will be stepping into the world of Artificial Intelligence. This paper will deal with an example of Natural Language Processing(NPL) and how a machine learning tool kit on python can be applied to text analysis. I would also like to talk about how this could could be used for more robust application in the future.

First let us talk abut unstructured data. The majority of data that the average person deals with is unstructured, “In today’s world of Big Data, most of the data that is created is unstructured with some estimates of it being more than 95% of all data generated.”(MongoDB, 2019) that is a lot of data that is MongoDB refers to as unstructured. But, we need to know what unstructured data refers to so we can get to the point of structuring so that we can manipulate the data for our own use. Emails, videos, pictures, audio, web pages, and social media messages are considered unstructured. We are unable to use that data to do anything else with it… unless we are able to. I will get more into this when talking about NPLs and artificial intelligence.

 So what is structured data?

I like to look at Margaret Rouse diagram of data structure hierarchy to get an idea of the depths that come with data structure. These are all ways to rearrange data so that a computer is able to easily interpret and use data. The first avenue I will follow is primitive data structure.

I like to think of primitive data as the first thing that anyone learns when programming. These are integers, floats, characters and Boolean operators. Integers are numbers that can be in any number system and easily stored by a computer. Integers start with a default value o 0. Floats are very similar to integers with two main differences. The first being that they are, for what I know, only represented in base 10 number system. And the second being that, they are able to hold decimal places. Characters and strings are letters and special characters that are able to represent actions. An example of a special character being \n that is a line feed that will act a the enter command in a text document, null is an empty string or character that can be helpful when coding (Oracle, 2019) . The last primitive data structure are Boolean operators. These operator are logical gates within software.

I do not want this paper to be incredibly long winded. I will talk about the subcategories of non-primitive data structures as two blanket ideas of linear and non-linear, and my dealings with each of them. The first is linear data. I think of this when dealing with scientific data. When I am doing astrophysics scrips, almost all of our data is arranged in a linear method. We need to know exactly what we are manipulating and how it impacts the output. There should not be a “black box” when doing physics. Everything is documented and understood, or at the least be theorized when getting a result. When I think of a non-linear data structure, I think of a “black box” or at least the idea and a black box being part of its functionality. My first thought is a hash table. The user has an input then it goes into a black box with some algorithm function doing thing to output something that lacks meaning without the algorithm.

Another non-linear data structure example is NPL. A cool toolkit that I have seen in the world of python is NLTK or the natural language toolkit. After reading the text for the package I quickly hurt myself in confusion, but I believe that I took away that the underlying structure of NLTK are trees that connect nodes together. There seems to be an algorithm that first defines what tree that It needs to be placed in. This process seems to be called a forest, but I that term seems to add a layer of ambiguity to an already complex idea. One is able to import text from many places and essentially manipulate it is which ever way you would like.

There are common things that text analysis tools are already used for. For example, as I write this paper I am getting suggestions of what I might want to say next. All I have to do is hit enter and it completes the text for me. Our smart phones are another great example. They will look back at words you have or have not used before and make or not make suggestions accordingly. The cool security hacker side of this is that programs can actually identify a person based on their selected words, punctuation, and sentence length. Another application that could be used is that someone could be identified by the manner in which they type. How the key strokes change in between letters or spaces, it software would be able to pick up on the rhythm that you use when typing! It is all very exciting.

I hope that this paper, if nothing else, was able to touch on the each Data structure and leave the reader with a entry level understanding of every one. Applying knowledge of data structures takes practice, and that cant be taught by reading. That requires Visual Studios Code with the python extension. I will be applying newly learned data structured by downloading The NLTK and playing with it a bit.

Works Cited

Rouse, Margaret. “Data Structure Hierarchy.” *What Is Data Structure?*, June 2019, cdn.ttgtmedia.com/rms/onlineimages/whatis-data\_structure\_desktop.png.

Oracle. “Primitive Data Types.” *Primitive Data Types (The Java™ Tutorials > Learning the Java Language > Language Basics)*, 2019, docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html.

“Unstructured Data In Big Data.” *MongoDB*, 2019, www.mongodb.com/scale/unstructured- data-in-big- data.