CMPSC 412 – Lab-10 (100 points)
Develop your own Data Structure

Due date: 12/09/2022

Lab Exercises:

Design and develop your own data structure to implement an application of your interest.

You can develop your own abstract data structure or implement a data structure from combination of the following data structures:

List, dictionary, sets, array, stack, queue, deque, tree, graph, linked list, heaps, priority queue and hash tables.

Your report should include the following:

1. **Introduction about the application:**
   context: I am traning my own voice AI, and I need a sizable dataset to make it work, and the dataset also need to be in a very specific format, doing the data gather manually will take a long time so I uses multiple programs and write multiple scripts to semi automate this process and this program part of that, it is needed for me to parse the data I've gather into the format that the AI can use to train.

   This applications uses command line arguments and it have three input, a text file with a specific format and an audio file and output directory, the text files is a subtitles file (.srt), the output directory is use to determined where to output all of the files this program generate, and it will also use a 3rd party program called FFMPEG to cut the audio files into multiple differents files.

2. **Design of the data structure:**
   The data structure will compose of 2 strings, and one array
   - name: name of the segment file
   - transcribe: the word spoken of the subtitles file.
   - time: an array consist of two time stamp, the timestamps are used to cut the files into multiple segments.

3. **Justification of developing the data structure:**
   My goal with this program is to extract the audio of an audio file into multiple segments, those segment would then be line up with what is being said in the subtitle files (.srt)

4. **Implementation and results:**

   the data class have 2 method, getMetaString and getFFmpegArgs, the two methods return strings and dictionary in a very specific format that can then be use to call FFmpeg.

   As a result this data structure play very nicely and I have successfully create a program that take in srt and audio files, then parse and cut it into multiple segments.

5. **Conclusion:**

   This program will be very useful for me even if I don't need it for the class anymore because it actually have usability that I need.

   I also understand that there are multiple approaches to creating a data type that fit one single need, currently my data type works like a container that store three pieces of data that I need, but I can also implement it further because of how I use it, in my program the data type is stored in an additional list, so instead of making the data type store 3 type of data I can also change it so that the data type stores a 3 list of 3 different type of data.

Deliverables:

Report and the program file.

You should demonstrate this application in class.