Ivan Ruan  
CISC 4900  
Prof. Huggins  
2/10/21  
 **Project Proposal Draft**

**Problem Statement** Accurate and in-depth knowledge in data structures is an essential component of a quality programmer. Recruiters that hire software developers quite often test prospective employees on this topic, and for good reason. Choosing the wrong type of data structure for a task will lead to end products that run extremely inefficiently or sometimes, not at all. As such, tech companies always make sure that their developers have a firm grasp on this facet of computer science. However, newcomers to the field of programming may find it difficult to remember or even learn when a type of data structure should be used. Even experienced programmers may easily forget. There are many options and the advantages and disadvantages of each are complex. Therefore, an app that can concisely convey this knowledge is needed. **Project Details** This application will display the qualities of various data structures in an easy to understand form. Users will be able to go through each and view their pros and cons. They will also be able to see real world uses of each to get a better idea of when each type of structure should be used. The app will also run find, insert, sort, delete on each structure, record their time and space efficiencies, and graph them. Finally, users will be able to take a short quiz in order to test what they’ve learned.   
 The technologies that will be used to develop this app are quite simple. The back end will be done in Java, while the front end will be done in Javafx and CSS using Scene Builder. The quiz feature will be implemented using a database of questions in H2 and connected with JDBC.  
 Although the tools used to build this project are few, I still have to self-teach myself quite a bit. I am not very familiar with Javafx and Scene Builder as I have only used fx a couple times in previous courses. At the time, I did not use Scene Builder and wrote the front end as part of the .java file. The GUI’s I made back then were quite simple and only spanned a single window. Therefore, I will have to look up tutorials on Scene Builder and learn how to create a well-designed GUI that allows for multiple scenes and graphs. As for the back end, although I am not using any unfamiliar technologies, the features of this app require functions that I have never used before. For example, I will need to look up how to measure space complexity as a function is running.

**Tentative Development Schedule**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| Gathering specs, put together project proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Create GUI scenes for each data structure and home |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Demo GUI to supervisor. Make any changes necessary |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Interim Report |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Create graphing feature |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test graphing feature and demo to supervisor |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Create quiz feature |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Test quiz feature and demo to supervisor |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Prepare final report and submit |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |