CSC 391/491
Programming Assignment 1
Algorithm Viewer

Due Date & Submission

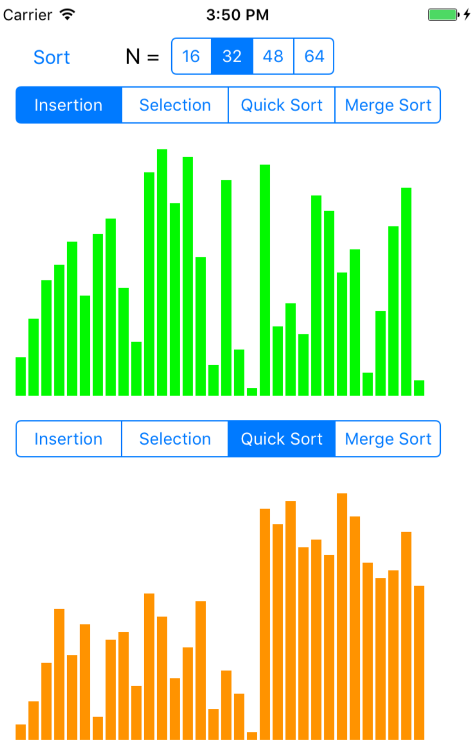
- Assignment due in 2 weeks.
- Submit your assignment in D2L Dropbox.
- Submit a single zip file that contains the contents of the project folder
    - To zip your project folder: Ctrl-click your project folder and select "Compress …" from the context menu.
    - **It is mandatory to use Xcode 15x and Storyboard for this assignment. No SwiftUI allowed!**
- Include only your source code files, including
    - *.swift, *.plist, *.xib, *.storyboard
    - image files
    - project files (.xcodeproj)
    - test folders
    - Screenshot
- You must use a unique prefix for the project name. (I suggest you use your last name and first initial as your prefix.) Please use the same prefix for all your assignments.
    - Note you only need to use the prefix for the project name. It is not necessary to use the prefix on other files in your project.
- Do not include unused or unrelated files.
- Before you submit, build and run the project, make sure everything compiles and works. Close your project before zipping the folder.
- Here are the most common reasons assignments are marked down:
    - Project does not build.
    - Project does not build without warnings.
    - One or more items in the Requirements section were not satisfied.
    - A fundamental concept was not understood.
    - Code is sloppy and hard to read (e.g. indentation is not consistent, etc.).
    - Your solution is difficult (or impossible) for someone reading the code to understand due to lack of comments, poor variable/method names, poor solution structure, etc.

Goals

- Building an app that uses the GCD for multi-threading.
- Following the best practices in memory management to ensure no cyclic strong references.

Requirements

- Develop an iOS app that animates various sorting algorithms. A sample UI design is shown in Figure 1. The two views can animate two different algorithms simultaneously using the same data.



- The specific requirements are:
  - UI:
    - A Sort button that starts the sorting process.
      - On tap, disable the Sort button, sample picker, and the 2 algorithm pickers.
      - On tap, both charts start the sorting algorithm, based on the selected algorithm and sample size.
      - **On completion of sorting, all elements should be reenabled.**
    - A Sample picker with options to pick array sizes: [16, 32, 48, 64]
      - On selection of sample sizes, each viewer should now have a different shuffled array with the selected sample size of elements.
    - 2 algorithm pickers and 2 chart views.

  - Animate the progress of the sorting algorithms by displaying the state of the array being sorted in a fashion similar to what is in the sample UI. Add small delays in the algorithms so that the progress is observable at a comfortable pace.

- Use a dispatch queue to execute the sorting algorithms on a thread other than the main thread.
- **Figure 1** A Sample UI Design You will need to create a custom view class to display and animate the state of the array being sorted.

Documentation

Possible bonus points for this assignment in one or more of the following areas:

- Exceptional quality in the design of your code, e.g., using suitable design patterns, language features, etc. to achieve simplicity.

- Properly handle all possible scenarios that may arise during the execution of the app.

- Clean, well-organized, and easy to understand code.
  Document the areas that exhibit these exceptional qualities in your app.

Troubleshooting

- If your project doesn't build and run correctly,
  - Make sure that iOS SDK and Xcode is installed and working on the machine
  - Verify that you have **iPhone Simulator** selected

Submission

1) A zip file containing your code.
2) A link to your GitHub repo.
3) A screenshot of your app.
4) **Make sure each of the above are individually submitted. (Do not package all three in one zip file).**