Mini Manual for:

Student: Sean O’Beirne

Date: 12/06/21

Course: CSCI 471 -01

Table of Contents

Table of Contents

Mini Manual

Objective:

Given this Training Manual and the sorter.py python file, the trainee will be able to visually display the execution of four different sorting algorithms including quick sort, heap sort, merge sort, and bubble sort.

Criterion:

The trainees will be able to configure Sorter to their liking, and from there they may run whichever sorting algorithm they like. They will then be able to reset Sorter, reconfigure if desired, then once again choose a sorting algorithm to run.

Target Population Description

Any person who is searching for a deeper understanding of these 4 sorting algorithms should be able to gain some level of such understanding from this manual.

Task Listing:

Using the software consists of:

- Launch Sorter
- Configure the number of bars on the display
- Execute a sorting algorithms
- Reset Sorter
- Reconfigure Sorter, if desired
- Execute another sorting algorithm and repeat

Training Module

You will use this training module to understand how to interact with Sorter. Specifically, you will be able to run one of 4 different sorting algorithms on a list of different-length bars.

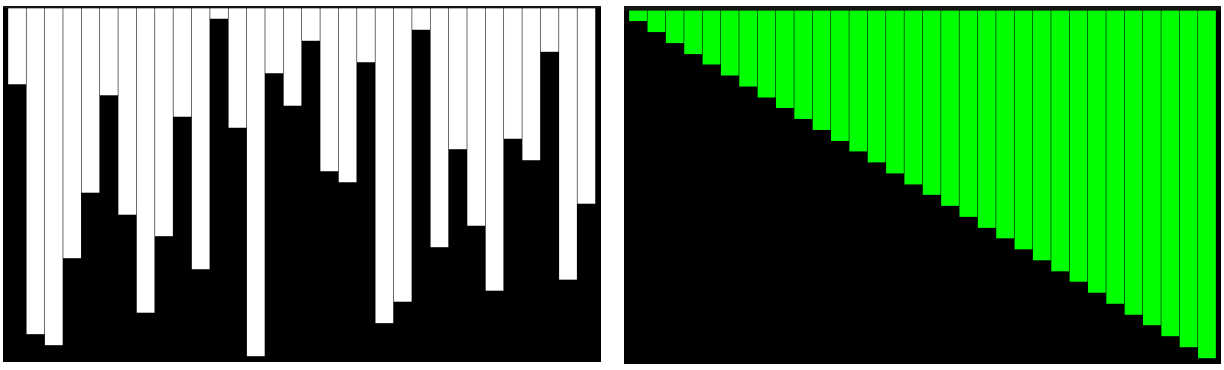
Software Description

Introduction

Imagine you have a list of batting averages for the top 100 baseball players of all time. This list of players is currently ordered by the year in which the player retired from their major league team. Certainly, this data would be interesting and useful to some extent, but say you wanted to find the top 5 batting averages of all time, and compare them against one another to see how your favorite players stack up against each other. How would you do this? Well, you'd have to look through the entire list, find the number one player, write his stats down, then find the number two player, and so on. This would get quite tiresome, especially if you wanted to get the top 50! So instead, you choose to sort the list by batting average in descending order first, then viola! You have your top 5 or top 50 right at the top! Well, maybe you could sort the list by hand for a list of 100, but how about for every baseball player of all time? I don't think so… Instead, the community of computer scientists have developed different algorithms to accomplish this exact type of task. Sorter is simply an application that is meant to visualize for you the steps that a computer might go through to sort such a list.

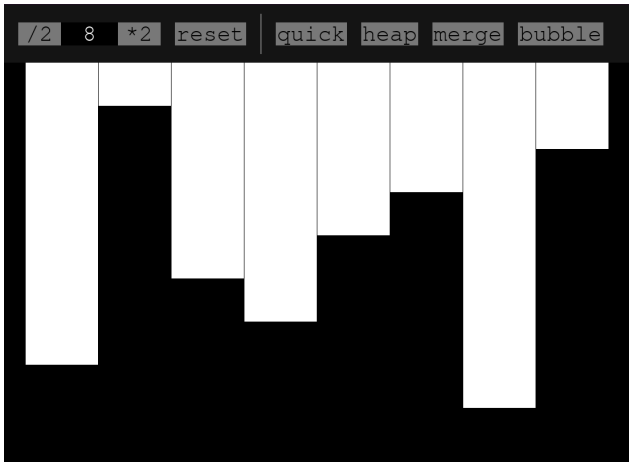
Background

Sorter does not show you a list of numbers that it sorts, as this may be quite hard to easily visualize as items are moved and swapped with one another. Instead, Sorter uses bars of unique lengths to represent different numbers so it's easy to follow how items are moving, and it's easy to know when different items are sorted. See the below graphic and take note of how clear it is that the screenshot on the left is scrambled in a random order while the screenshot on the left is clearly sorted in ascending order! This is what makes Sorter a useful application and education tool. Note that the color also changes to green when each item is sorted! How convenient!



Human Process Description

This screenshot shows what Sorter will look like upon the initial running of Sorter, which can be accomplished by running “python3 sorter.py” in any environment which has both python3 and the ‘pygame’ python3 package installed.
Sorter has two major components, the UI which is found at the top of the window, and the display, which consists of the scrambled array of bars below the UI. These bars will be in a random order between each running of Sorter, but there will always be 8 bars present to start.



The UI element of Sorter is what we will be conducting training on in this manual. Outside the scope of this manual would be understanding how each sorting algorithm works and understanding the color-coding of the bars as the algorithms are run. Atop the window in the UI element (indicated by a dark gray background) we will see 7 light gray buttons and 1 display field, which tells us how many bars are currently on the screen. There is a clear split between the first 3 and last 4 buttons, indicated by a thin light gray bar. On the left side of the bar, we can configure sorter, and on the right side, we can run our sorting algorithms.

The first step is to configure Sorter, which is done by clicking the ‘/2’ and ‘*2’ buttons on the left. This configures sorter by changing how many bars are on thee screen. There can be at minimum 2 and at maximum 512 bars on the screen, with increments of powers of 2. From here, we can click ‘reset’ to scramble the bars, or we can click on any of the 4 sorting algorithms on the right of the bar. Simply clicking on one of the buttons is enough to execute a sorting algorithm, and you will immediately see the sorting begin. Once the bars are in order, you may once again scramble the bars with ‘reset’, change the number of bars with ‘/2’ and ‘*2’, or run another sorting algorithm. You may repeat this process indefinitely! Do note though, that you may not change the number of bars on the screen mid-sort, as this would totally derail the sorting process. Similarly, you are not able to click on one sorting algorithm as a sorting algorithm is running. You may; however, stop the sorting and scramble the bars by clicking ‘reset’ at any time.

And with that, you are ready to use Sorter!