Lab 9: Analysis of Sorting Algorithms

CS134C (Spring 2019) - due Thursday/Friday at noon

In this lab we're going to experiment with Jupyter Notebooks. According to the Jupyter Notebooks website, "The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more." It's a very popular approach for scientists, using python, to document their data analysis process and share with others. You may have already encountered Jupyter Notebooks, as matplotlib uses one for its documentation.

Lab Requirements

For this lab, we're going to use Jupyter Notebooks to focus on the analysis of several sorting algorithms. You'll need to complete the following tasks:

1. Create a new cell at the top of this notebook, with (c) 2019 YOUR NAME and a brief description of this document.
2. Complete the brief descriptions under Our Sorting Algorithms.
3. Complete the _partition(d, low, high) function to modify the list, d, such that everything to the left of d[low] is less than d[low] and everything to the right is greater than d[low]. low is the index of the pivot value for one partition in quicksort. low is also the starting index for sorting, while high is the ending index for sorting. d may have a length of 10, but low may be 5 and high may be 8. Just partition the elements of d between 5 and 8 (inclusive).
4. We'll construct the shuffled(l) function together as a class.
5. This notebook provides a plot for three sorting algorithms applied to a sorted list. Add cells to the bottom of this notebook that also provide plots for performance on a reverse-sorted list, and plots for performance on a shuffled list (use the shuffled(l) function). Add additional cells, if you deem them necessary.
6. Under Discussion, provide a brief analysis of the differences you observe in algorithm run-times. Some thought questions are included to help guide your analysis.

Extra Credit

1. Write a new function, _randPartition(d, low, high), where d is a list to partition, low is the starting index for sorting/partitioning, and high is the ending index. Instead of always using the leftmost index (i.e., low) as pivot use a random index. Generate a graph of a revised Quicksort on a sorted list using this partitioning approach, and compare it to the performance of the Quicksort using the leftmost element as pivot.
2. Insertion Sort is often used for smaller lists, and Quicksort is used when the list gets bigger. Add cells as needed to this notebook to generate plots investigating the run-time for smaller lists, and explain why Insertion Sort might possibly replace Quicksort in some circumstances.
Set-up for this Lab

To do this lab, you're going to need to:

1. Open up your virtual environment from the cs134 directory via the terminal.
2. Install Jupyter: `pip install jupyter`
3. Install matplotlib (if you haven't already): `pip install matplotlib`
4. Clone the repository for this lab: `git clone ssh://22xyz22@davey.cs.williams.edu/cs134/22xyz22/lab9.git ~/cs134/lab9/`
5. Open the lab9 notebook in the lab9 directory: `jupyter notebook lab9.ipynb`
6. Place a new cell at the top of this notebook with your name, copyright year, and a brief description of what this document is.
   - Click the + button in the top left corner, and then use the scrollbar menu in the top middle to designate Code, Markdown, Or Raw

Be sure to save your work frequently, and `git commit -am "comment"` frequently, but in a different terminal than the one you're running your jupyter notebook in!

When You're Done

When you want to close the Jupyter Notebook, go to 'File > Close and Shutdown Notebook' in the Jupyter file menu.

Remember to finally add and commit your notebook before the end of lab.

Jupyter Notebook Structure

A Jupyter Notebook is composed of several cells. These cells can contain one of the following:

- blocks of python code
- blocks of raw, plain text
- blocks of a lightweight markup language for text formatting, known as Markdown. This lab notebook uses Markdown to make headers, show images, provide links, etc.

When you run the selected cells and advance via the play button at the top-left of the notebook, the notebook runs only the current cell. Not the ones before it, and not after. You can use run in the Jupyter File menu to Run All Cells. When there's output (such as for python code), it appears in an output cell after the input cell.

Note: if you run into issues with graphs not displaying, it's sometimes helpful to click the 'Restart the kernel' button in the top-left, but then you need to run each of your cells again!

Aside: How To Create a New Notebook

We're working from an existing Jupyter Notebook today, but if you wanted to create a new one for future use, you can do the following:

1. Run jupyter notebook in the terminal
2. Use the graphical user interface to navigate to Create New > Notebook > Python 3
3. Rename notebook by clicking Untitled next to jupyter logo at top-left.