

Introduction to Algorithms Programming Assignment 1

This programming assignment is due by midnight of 4/26 (Friday). Please submit the Python code with proper comments via eeclass. 20% penalty will be applied if it is submitted on 4/27. No submission will be accepted after 4/27.

This programming assignment allows you to visualize the execution of different sorting algorithms and compare their speed.

Sorting algorithms considered include **bubble sort**, **insertion sort**, **merge sort**, **quick sort**, and **heap sort**. (Please study heap sort, i.e., Chapter 6 of the textbook, by yourself.)

1. Generate a random permutation of 50 numbers ranging from 1 to 50.
2. Create a bounding box for each of the sorting algorithms listed above. Each bounding box contains 50 vertical lines with their lengths (from left to right) equal to the permutation in Step 1. At the end of Step 2, you should have 5 identical bounding boxes, each looks just like a histogram.
3. Conduct the respective sorting algorithm in each bounding box by rearranging the vertical lines **in parallel**. Each swap of two numbers is represented by the exchange of two vertical lines. A sorting algorithm in a bounding box is complete when its vertical lines are in ascending order of their lengths from left to right.

Hint 1: The Python animation can be done using matplotlib API.

Hint 2: You should create a big loop outside of all these sorting algorithms. In the big loop, each sorting algorithm will perform exactly one comparison of a pair of numbers. The big loop terminates when all sorting algorithms are complete.