Programming Assignment 1 (Due Friday, October 4, 2024)

Analysis of Algorithms

In your research project, you have collected a huge number N of data points $X_i, i = 1, 2, \ldots, N$, each of which is in the form of 2-D coordinates such that $X_i = (x_i, y_i)$, with x_i and y_i being floating point numbers with only 2 digits after the the decimal point. For example, a data point could be $(324.59, -100.08)$, but won't be $(6.981, -0.08)$, since in the latter case, the x-coordinate of the point is given to 3 digits after the decimal point.

You will also be given a separate data point $Z = (x_z, y_z)$, also given to two decimal points, at run time. Your job is sort all the N points $\{X_1, X_2, \ldots, X_N\}$ based on their Euclidean distance d_i to the point Z, where d_i is defined as

$$d_i = \sqrt{(x_i - x_z)^2 + (y_i - y_z)^2}\,.$$

and X_i's will be ordered in increased distance.

Your plan to use both Heapsort and Quicksort algorithms for this, and compare the runtime for different input sizes $n = 10^3, 10^5, 10^7, etc.$ For this, you will need to do the following:

1. Read the input data from an input file, which has the format of

```
z_x     z_y
N
x_1     y_1
x_2     y_2
x_3     y_3
x_4     y_4
.
.
.
x_N     y_N
```

   In other words, the input file contains $N+2$ lines, with the first line being a pair of floating point numbers representing the $x-$ and $y-$ coordinates of the point Z, respectively. The second line consists of a single integer specifying the number of data points to follow, and the subsequent N lines listing each data point, with the x- and y- coordinates separated by "Tab" and/or white spaces.

2. Implement both Heapsort and Quicksort. You will be given sample input files and expected sorted outputs for testing purpose.

3. Use the Heapsort and Quicksort algorithms you have implemented to sort the input array, and output two files, named "heapsorted_points.txt", and "quicksorted_points.txt", respectively, with both output files using the same format as the input file: 1) outputing the total number N of sorted points in the first line, and 2) outputing the sorted list of $X_1, X_2, \ldots, X_N$ in the subsequent N lines, one point per line.

4. Compare the runtime of your algorithms for input sizes $n = 10^3, 10^5, 10^7$. What is your algorithms' time complexity? Are they what you have expected?

5. If two points X_i and X_j are of the same distance to Z, how will your code behave? Is it the intended behavior?

6. **Bonus** If you have implemented any visualization method for your data during your implementation, which helps you in the implementation, debugging or confirming the correctness of your results, please include it in the submission and the writeup for bonus points.

What to Submit:

Please submit a write-up, and the source code to Blackboard before Friday, Oct. 4th, 2024. In the write-up, please provide a concise description of your method and rationale for design and implementation decisions. Please also provide answers to the requirement instructions above, if any.

Your write-up may consist of three types of content: *Analysis*, *Results*, and *Source Code.*

Analysis: This section should include relevant algorithmatic reasoning necessary to solve the problem or answer those particular questions.

Results: Output of your program and explanation of the results. Answers on qualitative and quantative questions. Discuss why certain methods did or did not perform as you have expected for different given situations, compare computed and theoretical results, etc. Use tables whenever appropriate. Bonus points will be given to optional graphics and plots.

Source Code: The source code should be readable and commented appropriately. Internal comments should describe algorithms and variables, relating them to those described in your Analysis section. Briefly describe the inputs and outputs of your code. Your program should be ready to run on a machine for verification purposes if needed for addressing concerns from the instructor or the TA.

Note: The programming assignments and associated write-ups must be done individually. However, discussing with your classmates or the instructor is encouraged.