

README – For DataSortingApp.jar

Purpose/Uses

- Provides an easy method of sorting large tables of data quickly, and provides several sort options that can help optimize the sorting process.

User Interface

- **Input/Output (IO) Tab** - This tab is the default tab, and sets the chosen files used as input and output.
 - o Input File and Output File - the text fields offer a direct method of inputting the file paths, as well as showing the path of any file chosen using the file chooser buttons to the right of each text field. The buttons provide a more familiar method of selecting files by using a file chooser to populate the file path text fields. Also note that using the File Chooser to select the input file will also set the output filename.
 - o The remaining two buttons labeled "Rank" and "Sort" gives the user options on how to organize the data. Ranking the data maintains the original order while sorting will change the order of the data.
- **Option Tab** - This tab provides many options that add to the program's versatility. These options mainly pertain to how it processes the data sets.
 - o Sorting Algorithm - This field is controlled by a combination box and offers different types of sorting that may or may not speed up the process of sorting, depending on the data itself. Information about the sorting algorithm can be seen on the right of the tab.
 - o Max. Columns and Max. Rows - These fields are self-explanatory, as they set the maximum number of columns and rows the program must process. This is useful if the user only wants certain data organized. The fields can be set directly, or by using the arrow buttons.
 - o Column to Sort by - This field sets the column of data to sort the data by. Note that this field is limited by the current setting on the Max. Columns field.
 - o Header-Related Checkboxes - These two checkboxes simply tell the processor whether there are headers in the original file, and whether it should output the sorted file with headers.
- **InfoLog Tab** - This tab acts as an information log that is used to communicate with the user, providing them with performance information and error messages.
- **DataTable Tab** - This tab allows the user to view the sorted result of the data without needing to open an external program to do so. This saves the user time and is more convenient.
 - o The table displays the data sorted with headers, so the user may identify each column of data. The column headers are not affected by the header output settings. The user has the ability to rearrange the columns so they may compare two non-adjacent columns easily.

FAQ

- The Information Log says that it took n time to sort my data, but it took a lot longer than n !
 - o The Sorting algorithms only calculate the time it takes to SORT the data. The extra time was most likely from the program attempting to render the data table.
- The program won't read my data!
 - o Make sure the data is in a .CSV format. In a CSV format, the data within rows is separated by commas, not tab characters like in .XLS and .XLSX format.
- How do I move, minimize, and close the window?
 - o You can move the window simply by clicking empty space on the outside border and dragging. The two buttons on the top right corner are (from left to right) the minimize button and the close button.
- Can you explain how the sorting algorithms work??
 - o Why don't you Google the names of the sorting algorithms?? All of them except the "Min-Max Sort" can be found online. The "Min-Max Sort" is essentially a selection sort which selects the least and greatest values and places them at the ends.
- Can I have your source code??
 - o Use the following Java code to generate the answer:

```

public class AnswerQuestion
{
    public static void main(String[] args)
    {
        String q = "Can I have your source code?";
        String a = "";
        int n = 0;
        for (int i = 0; i < q.length(); i++) n += (int) q.charAt(i);
        if (n%2 != 0) a = "Yes.";
        else a = "No.";
        System.out.println(a);
    }
}

```

- Explain what each of the terms mean in the sorting algorithm info box.
 - o The first line is the name
 - o The second line is the type. By type, it means whether or not the sort compares the data to each other to get the final result.
 - o "Method", refers to the way it sorts the data.
 - o "Stable?" refers to the whether the algorithm keeps the relative position between identical values.
 - o The best, average, and worst lines refer to the time complexity of the algorithm. Best = the quickest run-thru while worst = the slowest. Generally you want the worst-case to be low. For more information on this, search "time complexity".
 - o "Memory" refers to the amount of extra data it needs to save (aside from the data itself) to sort the data. The best algorithms in terms of memory are called in-place sorts, which require no additional space.
 - o "Strings?" simply refers to whether the programs implementation of the sort can handle text (non-numerical values) as the sorting values.