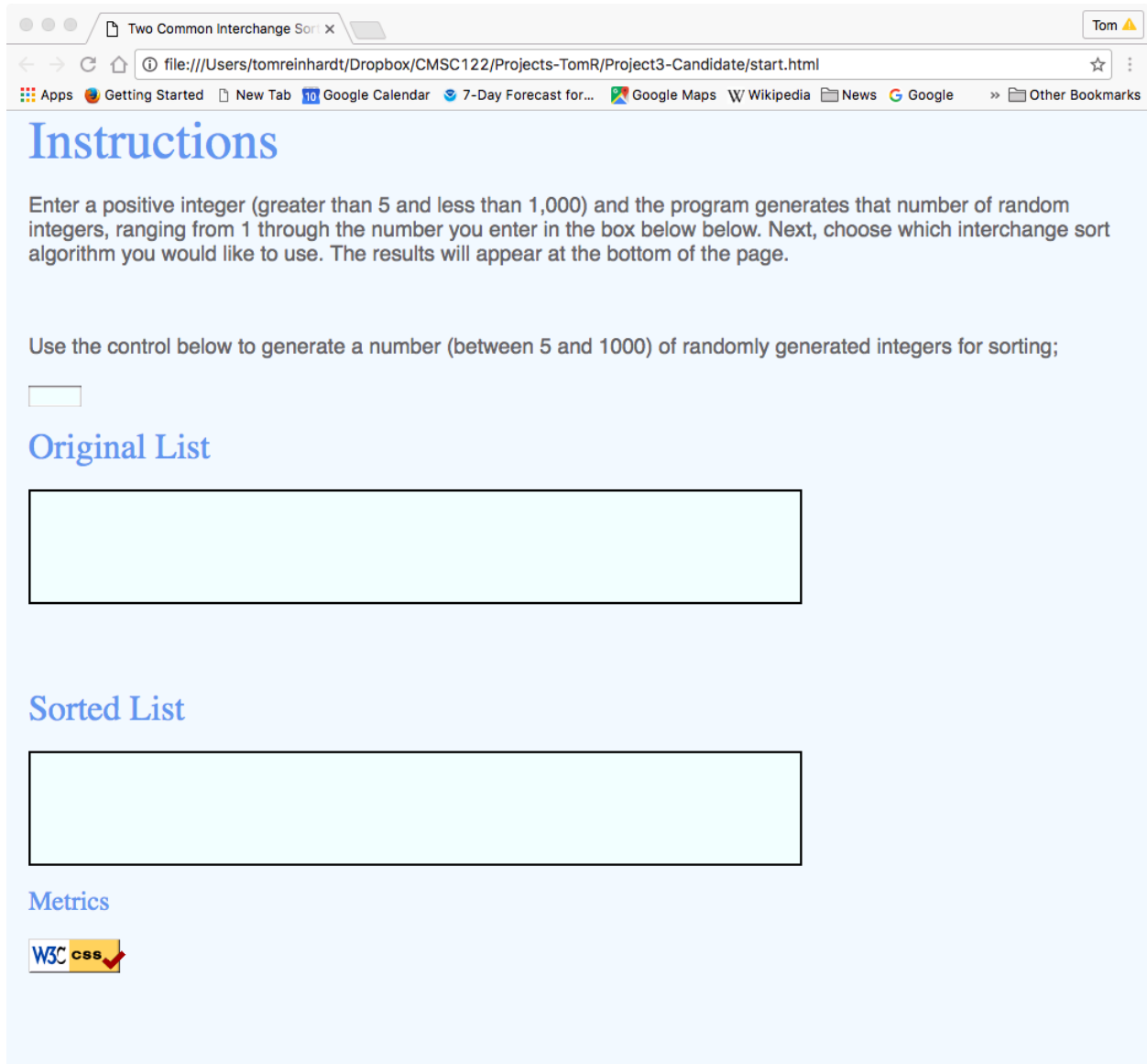


Interchange Sorts: Project 4

Instructions

For this project you will construct a web-page allows users to generate test data (lists of integers) and sort them by selection and/or insertion. The initial page should look like:



The screenshot shows a web browser window with the title 'Two Common Interchange Sorts'. The address bar shows the file path: `file:///Users/tomreinhardt/Dropbox/CMSC122/Projects-TomR/Project3-Candidate/start.html`. The browser's bookmark bar includes links to 'Apps', 'Getting Started', 'New Tab', 'Google Calendar', '7-Day Forecast for...', 'Google Maps', 'Wikipedia', 'News', 'Google', and 'Other Bookmarks'. The page content is as follows:

Instructions


Enter a positive integer (greater than 5 and less than 1,000) and the program generates that number of random integers, ranging from 1 through the number you enter in the box below. Next, choose which interchange sort algorithm you would like to use. The results will appear at the bottom of the page.

Use the control below to generate a number (between 5 and 1000) of randomly generated integers for sorting;

Original List

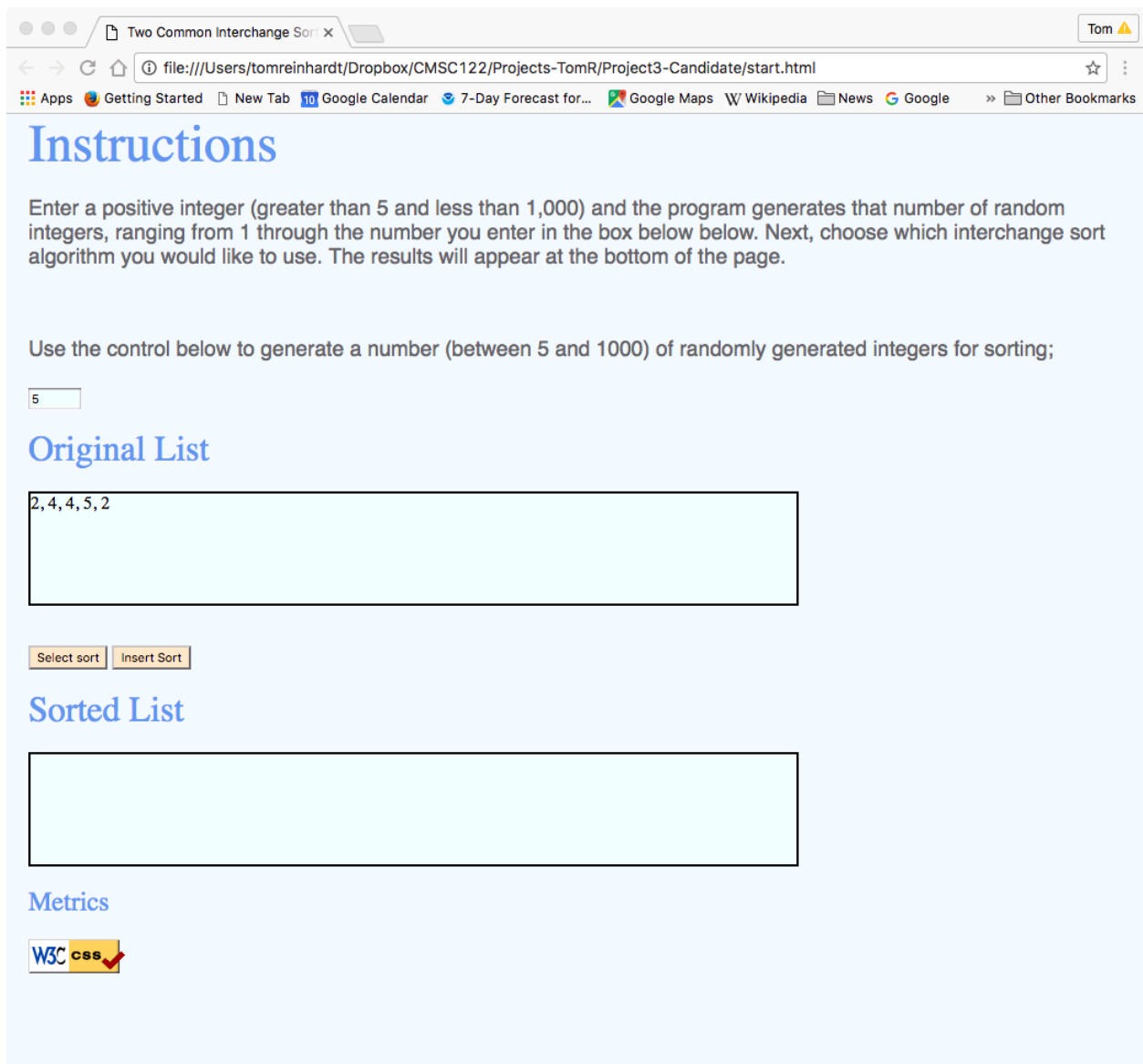
Sorted List

Metrics



Note: the image that appears at the bottom of the illustration above was created by the CSS Validator page. Upon validating your css file, you will be given some boilerplate HTML text that you should paste into the bottom of your HTML document.

Note also that NO sort options appear, until the user instructs the program to generate some test data, which is illustrated on the next screen shot:



The screenshot shows a web browser window with the title "Two Common Interchange Sort: X" and a user profile "Tom". The address bar shows the file path: `file:///Users/tomreinhardt/Dropbox/CMSC122/Projects-TomR/Project3-Candidate/start.html`. The browser's bookmark bar includes links to "Apps", "Getting Started", "New Tab", "Google Calendar", "7-Day Forecast for...", "Google Maps", "Wikipedia", "News", "Google", and "Other Bookmarks".

Instructions

Enter a positive integer (greater than 5 and less than 1,000) and the program generates that number of random integers, ranging from 1 through the number you enter in the box below. Next, choose which interchange sort algorithm you would like to use. The results will appear at the bottom of the page.


Use the control below to generate a number (between 5 and 1000) of randomly generated integers for sorting;

Original List

2, 4, 4, 5, 2

Sorted List

Metrics



The user can choose either the Select or the Insert sort routines, which will generate some sorted output and some statistics collected during the sort. This is illustrated on the next panel:

Two Common Interchange Sort x Tom

file:///Users/tomreinhardt/Dropbox/CMSC122/Projects-TomR/Project3-Candidate/start.html

Apps Getting Started New Tab 10 Google Calendar 7-Day Forecast for... Google Maps W Wikipedia News Google >> Other Bookmarks

Instructions

Enter a positive integer (greater than 5 and less than 1,000) and the program generates that number of random integers, ranging from 1 through the number you enter in the box below. Next, choose which interchange sort algorithm you would like to use. The results will appear at the bottom of the page.

Use the control below to generate a number (between 5 and 1000) of randomly generated integers for sorting;

5

Original List

2, 4, 4, 5, 2

Select sort Insert Sort

Sorted List

2, 2, 4, 4, 5

Metrics

Number of comparisons = 15 and the number of swaps = 2

W3C CSS

Note: The color shift on this illustration is an artifact of the program used to create the screen snapshot and is NOT representative of what your page should look like; your colors should remain the same throughout users interactions. The border around the “Select sort” button (or the “Insert sort” button), however, will have some border coloring effect.

The data that was created, i.e., the sorted list of integers and the “Metrics” are required.

Note: the user should be able to choose the other sort and the program should apply that algorithm to the original data and generate new metrics showing the number of comparisons and swaps required by the sort routine selected. The next couple of screen shots show the results of sorting 200 randomly generated integers: first using Selection sort, then using Insertion sort on the same data—this was done by selecting the desired sort (clicking on the “other” sort button) without re-generating the test data.

So, here's the result of sorting 200 randomly generated integers by a selection sort algorithm:

Two Common Interchange Sort x Tom

file:///Users/tomreinhardt/Dropbox/CMSC122/Projects-TomR/Project3-Candidate/start.html

Apps Getting Started New Tab Google Calendar 7-Day Forecast for... Google Maps Wikipedia News Google Other Bookmarks

Instructions

Enter a positive integer (greater than 5 and less than 1,000) and the program generates that number of random integers, ranging from 1 through the number you enter in the box below. Next, choose which interchange sort algorithm you would like to use. The results will appear at the bottom of the page.

Use the control below to generate a number (between 5 and 1000) of randomly generated integers for sorting;

Original List

54, 27, 185, 162, 146, 81, 170, 110, 140, 161, 1, 180, 50, 161, 146, 16, 59, 69, 103, 163, 37, 59, 98, 118, 69, 81, 184, 91, 169, 155, 129, 148, 163, 24, 17, 149, 145, 193, 70, 85, 78, 101, 116, 68, 26, 162, 13, 10, 116, 78, 14, 122, 140, 151, 120, 110, 16, 131, 30, 65, 118, 170, 184, 71, 168, 9, 42, 72, 134, 117, 140, 166, 20, 100, 108, 142, 7, 59, 127, 186, 59, 113, 157, 130, 179, 163, 35, 70, 77, 109, 152, 81, 3, 57, 28, 16, 155, 165, 84, 115, 71, 84, 191, 75, 174, 74, 56, 14, 153, 157, 122, 141, 34, 11, 78, 154, 103, 155, 3, 68, 142, 6, 64, 5, 80, 186, 109, 177, 110, 76, 169, 167, 147, 146, 145, 107, 47, 159, 89, 81, 122, 179, 44, 19, 123, 47, 50, 51

Select sort Insert Sort

Sorted List

1, 1, 3, 3, 5, 6, 7, 9, 9, 10, 11, 13, 14, 14, 16, 16, 16, 17, 19, 19, 20, 24, 25, 26, 26, 27, 27, 28, 30, 34, 35, 35, 37, 39, 42, 43, 44, 46, 47, 47, 50, 50, 51, 54, 54, 54, 56, 57, 57, 57, 59, 59, 59, 59, 60, 64, 65, 66, 66, 68, 68, 69, 69, 70, 70, 71, 71, 72, 72, 74, 75, 76, 77, 78, 78, 78, 78, 80, 80, 81, 81, 81, 81, 81, 84, 84, 85, 85, 89, 91, 98, 99, 100, 101, 103, 103, 107, 108, 109, 109, 109, 110, 110, 110, 111, 113, 114, 115, 115, 116, 116, 117, 117, 118, 118, 120, 122, 122, 122, 123, 127, 129, 130, 130, 131, 133, 134, 136, 140, 140, 140, 140, 141, 141, 142, 142, 145, 145, 146, 146, 146, 147, 147, 148, 149, 151, 152, 153, 154, 154, 155, 155, 155, 157

Metrics

Number of comparisons = 20100 and the number of swaps = 195



Now, just clicking on the “Insert sort” button will re-use the same test data but should generate both a correctly sorted list of integers and different Metrics. This is shown below:

Two Common Interchange Sort

file:///Users/tomreinhardt/Dropbox/CMSC122/Projects-TomR/Project3-Candidate/start.html

Apps Getting Started New Tab Google Calendar 7-Day Forecast for... Google Maps Wikipedia News Google Other Bookmarks

Instructions

Enter a positive integer (greater than 5 and less than 1,000) and the program generates that number of random integers, ranging from 1 through the number you enter in the box below. Next, choose which interchange sort algorithm you would like to use. The results will appear at the bottom of the page.

Use the control below to generate a number (between 5 and 1000) of randomly generated integers for sorting;

200

Original List

54, 27, 185, 162, 146, 81, 170, 110, 140, 161, 1, 180, 50, 161, 146, 16, 59, 69, 103, 163, 37, 59, 98, 118, 69, 81, 184, 91, 169, 155, 129, 148, 163, 24, 17, 149, 145, 193, 70, 85, 78, 101, 116, 68, 26, 162, 13, 10, 116, 78, 14, 122, 140, 151, 120, 110, 16, 131, 30, 65, 118, 170, 184, 71, 168, 9, 42, 72, 134, 117, 140, 166, 20, 100, 108, 142, 7, 59, 127, 186, 59, 113, 157, 130, 179, 163, 35, 70, 77, 109, 152, 81, 3, 57, 28, 16, 155, 165, 84, 115, 71, 84, 191, 75, 174, 74, 56, 14, 153, 157, 122, 141, 34, 11, 78, 154, 103, 155, 3, 68, 142, 6, 64, 5, 80, 186, 109, 177, 110, 76, 169, 167, 147, 146, 145, 107, 47, 159, 89, 81, 122, 179, 44, 19, 123, 47, 50, 51

Select sort Insert Sort

Sorted List

1, 1, 3, 3, 5, 6, 7, 9, 9, 10, 11, 13, 14, 14, 16, 16, 16, 17, 19, 19, 20, 24, 25, 26, 26, 27, 27, 28, 30, 34, 35, 35, 37, 39, 42, 43, 44, 46, 47, 47, 50, 50, 51, 54, 54, 54, 56, 57, 57, 57, 59, 59, 59, 59, 59, 60, 64, 65, 66, 66, 68, 68, 69, 69, 70, 70, 71, 71, 72, 72, 74, 75, 76, 77, 78, 78, 78, 78, 80, 80, 81, 81, 81, 81, 81, 84, 84, 85, 85, 89, 91, 98, 99, 100, 101, 103, 103, 107, 108, 109, 109, 109, 110, 110, 110, 111, 113, 114, 115, 115, 116, 116, 117, 117, 118, 118, 120, 122, 122, 122, 122, 123, 127, 129, 130, 130, 131, 133, 134, 136, 140, 140, 140, 140, 141, 141, 142, 142, 145, 145, 146, 146, 146, 147, 147, 148, 149, 151, 152, 153, 154, 154, 155, 155, 155, 157

Metrics

Number of comparisons = 9631 and the number of swaps = 196

W3C CSS

Generally: we should expect the metrics to be a little better for the “insertion sort,” given the same data.

Requirements

The following are critical to how you will be assessed for this assignment:

- Pay attention to the visibility of elements. Note that the data-boxes containing unsorted and sorted integers are invisible UNTIL the user sets a number of random integers to be randomly generated. At this time, the Original List contents are populated.
- Upon choosing either sort, the Sorted List text and the Metrics text are (re-)populated.
- Choosing another sort after this point should re-use the test data, apply the newly-chosen sort, and re-populate both the Sorted List and the Metrics text areas.
- Both the Original List and the Sorted List boxes are “scrollable,” meaning that if I move my mouse to the right side of the bounding boxes, scroll controls appear allowing me to scroll through the numbers in a natural way.
- Your JavaScript functions must correctly implement both the “selection sort” and the “insertion sort” algorithms. (Do not implement a bubble-sort or some other kind of sort here.)
- The “color scheme” used here is: background-color: #f0f8ff, h1,h2 and h3 colors are #6495ed, and the scroll box background color is #f0ffff. You are, of course, free to use your own color scheme; make sure that we can clearly read your page however!
- Naturally, your HTML must validate to HTML 5.0 and your CSS must validate (it should validate to CSS 3). Naturally, your JavaScript code should contain NO extraneous output or generate any bugs.
- All files should be documented!

Submission Requirements

Submit to Elms a tar file (directory) named as always: yourName-Project4.zip. This zip file should contain:

- your HTML file: `start.html`
- your CSS file: `styles.css`, and optionally
- your JavaScript file: `sortingScripts.js`