1. Merge the two binomial heaps below.

![Binomial Heaps](image)

**Objective of Sort Detective**

The primary objective of this assignment is for you to apply your theoretical knowledge of sorting algorithms to solve a problem. More specifically, you will be given a program which is designed to measure comparisons and data movements for six sorting algorithms discussed in class. Unfortunately, the designer of the program did not label the buttons properly. You must apply your understanding of the general properties of the algorithms (and in some cases of the code used to implement them) to determine the proper labeling of the buttons.

The secondary objective of this lab is for you to gain experience writing a concise, but complete analysis of a system.

**Background**

As you know from class, if you double the size of the data set that you give to a quadratic algorithm, it does four times the work; by contrast, an $O(n \log n)$ algorithm does a bit more than twice as much; and a linear algorithm does only twice as much work. As you also know, the characteristics of the input data set can affect the expected performance of many of our sorting algorithms. Before you begin the lab, you should review the expected performance of the algorithms on various data sets.

The sorting algorithms under study include BubbleSort, HeapSort, InsertionSort, MergeSort, QuickSort, SelectionSort, and ShellSort.

**Instructions – Warning: read all of the instructions before beginning!**

To get to the web page for Sort Detective do the following:

Go to [csilm.usu.edu](http://csilm.usu.edu) in your browser. Under “CS3” select Sorting. SortDetective is the 5th activity. Work through the various options, comparing the various sorting methods. Note that the initial order of the data changes the run time. This is important to understand when we give a big O analysis and motivates our using an upper bound (rather than an average case).
1. Try various combinations of sorting algorithms and data type/size. Notice that the button names (Sort A-G) do not give any indication which sort they execute. The seven sorts represented are: Quicksort, Selection, Heapsort, Insertion, Shell, Mergesort, and Bubble sort. Notice also, that you need to create an array before you can sort it. There is a ‘Data’ button that generates an array with different characteristics and a “Size” button that generates different sized arrays. Important characteristics of sorts include: stability, adaptive, run time, and amount of data moved. While we can't observe the amount of data being moved, the other three characteristics can be observed. Ask yourself, how would you determine if a sort was stable, adaptive, or how you would determine complexity. The "Analyze" option of "Runtime" shows the runtime for various sized arrays.

2. Devise a plan which enables you to match the particular algorithms to the button names. Hint: It may make sense to try to divide the sorts into initial groups (based on features) and then to work on each group separately. Storing the counts in a spreadsheet may help you organize your attempts. Divide and conquer: it works for algorithms and it can work here, too!

3. List the questions you intend to answer in an appropriate order. Execute your plan, taking careful notes as you go.

4. Describe the results of your experiment in a summary document. Begin by listing which sort is associated with each of A-G. Then show the rationalization process that justifies your decisions.

A Note on Writing

A significant portion of the grade for this assignment will be determined by the quality of the writing of the report. This includes the completeness of the report, the clarity (and grammar) of the writing, and general presentation.

Some of the sorts are very difficult to distinguish. A carefully outlined experiment may compensate for an error in these cases if the writing makes it clear that your conclusions/guesses are substantiated by the data.

Finally, remember that your report needn’t detail every experiment you ran. Rather, it should give sufficient information to justify your conclusions. It is possible to write a very short report that is completely correct if your experiments are well-chosen. After you learn the matching, you might consider whether there was a shorter way to arrive at your conclusion!

Be aware that the work reported by the ILM is counted loosely – so while counts within the same sort are comparable, work counts between sorts are not.

Notes

- When completed, turn in your written assignment using eagle.cs.usu.edu in one of the following formats: .doc, .odp, .pdf or submit a printed copy of your assignment in class.