Week 3 Lecture 8

Theory

What’s in this lecture?

- Merge Sort
- Thinking about the Efficiency of Algorithms

Sorting

- Last time: explored two different sort algorithms
- Today: one more sort algorithm, and some quick precursors to algorithm analysis

Previous Approach

- Process entire list at a time, repeatedly removing inversions based on comparisons

Today's Approach

- Divide & Conquer!
- Turn an array into two arrays that are half the size, sort those, and then merge the results
- Insight: will not have to traverse the entire array repeatedly as in previous approaches

Merge Sort

```
function merge_sort(a_array) {
  if (a_array.length < 2) {
    return a_array;
  }
  var mid = parseInt(a_array.length / 2);
  var left = a_array.slice(0, mid);
  var right = a_array.slice(mid, a_array.length);

  return merge(merge_sort(left), merge_sort(right));
}
```

Merge Sort

```
function merge(left, right) {
  var merged = new Array[];
  while (left.length && right.length) {
    if (left[0] <= right[0]) {
      merged.push(left.shift());
    } else {
      merged.push(right.shift());
    }
  }
  while (left.length) {
    merged.push(left.shift());
  }
  while (right.length) {
    merged.push(right.shift());
  }
  return merged;
}
```

Thinking...

- How many times will merge_sort() get called for an array of size N?
- How many times will merge() get called for an array of size N?
- How big are the subarray arguments for each of these calls?
- Hint: you can measure this with console.log for some small N!

Looking Back...

- How many comparisons / swaps will bubble_sort do on an array of size N?
- How many comparisons / swaps will insertion_sort do on an array of size N?

Next time:

- More formally examine the running times of all of these algorithms
- Talk about the basics of algorithm analysis

Exercises

- Read Intro to Algorithms, 3rd Edition, Chapters 3 & 4 (lightly)
- Modify the sorting function to reverse the numeric sort order
- Make it so that the sorting function takes in a first-class compare(a,b) function that *you* write
- Instrument the three sorting algorithms with console.log calls to start getting a feel for running time