WRITE UP

Time Complexities

Binary Insertion Sort
Worst = O(nlog(n))
Avg = O(nlog(n))

Quick Sort
Worst = O(n^2)
Avg = O(nlog(n))

Shell Sort
Worst = depends on gap
Avg = depends on gap

Bubble Sort
Worst = O(n^2)
Avg = O(n^2)

What I learned/Experiments

This assignment has taught me a lot of things I hadn't ever really considered when it came to efficiency. One of my “oh this makes a lot of sense” moments was when I was testing my code and I noticed that the Bubble Sort method had a constant number of comparisons no matter what numbers were in the array. At that point it began to click that different sort functions each have their own way of trying to take as little steps as possible to solve the problem, and that the Bubble Sort was just a baseline example of a really easy to read sort, but a really inefficient one. After I had finally managed to get all of my sorts working I began playing around with them (and freezing my computer at one point) and noticed the vast differences in all of the sorts.

From my understanding, Bubble Sort is consistently the worst. It doesn't matter whether your list was almost sorted or not, it will always check every element in the array, and it consistently had the worst moves, almost tied with the Binary Insertion Sort. I guess that makes sense since both essentially just sift through the array, but at least the Binary Insertion sort managed to cut down the number of comparisons it has to make, which decreased its time complexity.

One of the things I wanted to do was declare a clear champion. The sort that works the best. On paper, it seems that the winner for time complexity is the Shell Sort. According to my own testing though, the Shell Sort could rarely ever hold a light to the Quick Sort, which actually has a *worse* complexity then Shell sort. Consistently the Quick Sort managed to pull ahead with the least numbers of moves and comparisons almost every time, which at the time didn't make sense to me, but after doing some reading does. Although on paper Quick Sort is slower, usually quick sort is much faster because its average complexity is low compared to its worst complexity. This means that in many cases, quick sort is actually faster than seemingly better methods. So while on paper it's the Shell Sort that wins, I declare Quick Sort the people's champion.