**Title Slide**

Hi, my name is Issam Ahmed and Ill be presenting my research in Comparison of sorting Algorithms.

**Introduction Slide**

For this research project I compared two main sorting methods, Iterative sort and recursive Sort.

Iterative sort uses loops in their algorithm to sort the data. There are three iterative sorts being used for this experiment. bubble sort , selection sort and insertion sort. These are a brief over view of how they each work.

For Recursive sort methods, it follows a divide and conquer method. It breaks down the data sets into smaller pieces to make it easier to sort. The two methods used here are the quick sort and merge sort.

Even though recursive methods are more complex and have potential overhead issues, they have much faster processing time compared to iterative sorting methods. The recursive method will outperform iterative sort with any data set size.

**Argument Slide**

Why is that? Well its mostly do to the structure and methodology of the algorithm. Lets first look at the temporal complexity. Since Iterative sort uses nested loops to sort data, it has a high temporal complexity. In fact the average performance is O(n2), give a quadratic relationship. For whatever data size, the processing time will grow exponentially.

Recursive sort breaks data into smaller and easier sets to compare. The average performance is O(n log n). With this type of relationship, the processing time for any data set size is much less than iterative. Larger data sets will only have a small impact on time.

Let’s look at the spatial complexity for the sort methods. Iterative has very little overhead but at the minimum must go thru the list for each element it has to sort. Therefore, it has a linear spatial complexity, the processing time grows about the same as the growth of the data set. The problem with recursive methods, is the process has recursive overhead. It creates new stack frames each time the method is called recursively. However, it uses a depth first method, so it goes down one branch of the data and deletes the stack frames as it back out. This only creates a very little overhead, which is neglectable. It also has a logarithmic spatial complexity.

Therefore, due to both of these complexities, recursive is a more complex algorithm but much faster than iterative.

**Test Slide**

To test this, I created a program that run all five methods with array of randomized 6-digit numbers and let each method sort it. I used 7 data sets, sizes ranging from 10 thousand to 20 million elements. Each data set size was ran 5 times for each method, with a completely new array each time. Any sort that took more than 5 minutes was manually terminated. If it failed at a lower data set, it was not repeated at a higher data set. For this experiment I kept an equal system environment at 10% CPU usage.

These are the result of the tests. From the table it is shows the average processing time for each method and data set. This shows that recursive methods outperformed the iterative methods for all data sets. The blanks show where the sort methods took more than 5mins and was terminated.

Here is a graph to summarize the data gathered. The bubble sort took the longest for each data set and failed after the 200 thousand set. Selection and insertion sort performed a bit better however, failed after the 1mil data set capping at average 163 seconds and 250 seconds. The growth for these iterative sorts is exponential. Quick and merge sort however, had only a slight change in processing time, only taking about 2.5 seconds to sort a 20million data set.

**Conclusion Slide**

As seen from the results, the recursive sorting methods outperformed the iterative methods with all data sets. This is again because the difference is their temporal and spatial complexity. For iterative sort, larger data equals an exponential growth in time, and for recursive larger data only cause a small growth in processing time. The recursive method also did not crash or have errors, thus showing the recursive overhead is neglectable.

So in conclusion, quick sort and merge sort is far superior. Even though they are more complex algorithms, once learnt they can be easily implemented. Bubble sort, Selection sort and insertion sort are a good introductory algorithm for learning sorting but its not recommended in real world application. For further tests it would be interesting to try it in different environments and different types of data like string, but the results would be the same. Recursive Sort methods will have a faster processing time than iterative sorts methods with any data size.

**For more information Slide**

For more information, check out these websites and videos to get a clearer understanding of different sort methods.