**Project 1**

**Hybrid and Adaptive Sorting Algorithms**

Tyler McCluskey

**PROJECT GOALS**

In this project, students will explore and implement hybrid or adaptive sorting algorithms to improve the efficiency of the sorting process through piggybacking the existing sorting algorithms.

**ALGORITHM DESCRIPTION**

The algorithm I went with improving is hybrid quicksort. This algorithm combines both quicksort and insertion sort. Insertion sort is better when the array is small, so the algorithm uses insertion sort when the array is small enough. Otherwise, it uses quicksort.

How I improved this algorithm is firstly I improved the insertion sort by making adding making use a binary search within the sorted portion of the array to find the correct position to determine where to insert each element. This reduces the number of comparisons compared to a standard insertion sort. I improved the quicksort by making it a randomized pivot instead of just the highest value.

Altogether my algorithm consistently performs better than the traditional hybrid quicksort. I call it Hybrid Adaptive Quick Sort as it uses elements from both a hybrid and an adaptive sorting algorithm.

**BENCHMARKING RESULTS**

-----------------------TEST 1-----------------------

Best Case hybrid quick adaptive sort execution time: 0.001001119613647461

Worst Case hybrid quick adaptive sort execution time: 0.001001119613647461

Best Case hybrid quick sort execution time: 0.002996683120727539

Worst case hybrid quick sort execution time: 0.0029985904693603516

-----------------------TEST 2-----------------------

Best Case hybrid quick adaptive sort execution time: 0.0009965896606445312

Worst Case hybrid quick adaptive sort execution time: 0.0

Best Case hybrid quick sort execution time: 0.0029990673065185547

Worst case hybrid quick sort execution time: 0.003001689910888672

-----------------------TEST 3-----------------------

Best Case hybrid quick adaptive sort execution time: 0.001001119613647461

Worst Case hybrid quick adaptive sort execution time: 0.0009989738464355469

Best Case hybrid quick sort execution time: 0.003000497817993164

Worst case hybrid quick sort execution time: 0.0029993057250976562

-----------------------TEST 4-----------------------

Best Case hybrid quick adaptive sort execution time: 0.0

Worst Case hybrid quick adaptive sort execution time: 0.0010023117065429688

Best Case hybrid quick sort execution time: 0.0029981136322021484

Worst case hybrid quick sort execution time: 0.0029976367950439453