Name(s)

Jaston Epp; Samuel Taylor

Project Number

S1406

Project Title

Sorting Algorithms

Abstract

Objectives/Goals

The objective of this experiment is to find which sorting algorithm is the fastest from five common algorithms. Our hypothesis was that mergesort would be the fastest as it takes advantage of it being easier to sort a short list than a long list and is able to break down the complexities of sorting thousands of numbers into many simple operations of comparing only two numbers.

Methods/Materials

In order to test our hypothesis, my partner and I created a program in Python that generates a list of 1,000 random integers in no particular order and then measures how fast each algorithm is able to sort that list from least to greatest. Materials include a computer running Windows and Python.

Results

In our 1,000 tests, the average sorting time for insertion sort was 0.000505785 seconds. The average sorting time for mergesort was 0.004430423 seconds. The average sorting time for heap sort was 0.005519048 seconds.
The average sorting time for quicksort was 0.05777026 seconds. The average sorting time for Bubblesort was 0.106378149 seconds.

Conclusions/Discussion

Insertion sort turned out to be the fastest sorting algorithm. We believe this may be because of the extra time mergesort must have to merge lists.

Summary Statement

The project's purpose is to find the fastest sorting algorithm out of five common algorithms.

Help Received

CodeCodex was consulted to aid in writing the program. We learned Python from MIT OpenCourseWare.