ME 5773

Team 3

Team Members:

Samuel Roberts

James Standard

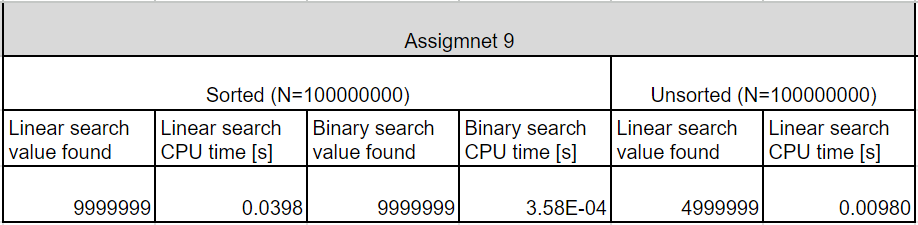
Homework Assignment 9 & 10

**Problem 1**

Searching algorithms are essential for many numerical applications. Given an array, and a point of interests, the goal of the search algorithm is to identify where the point of interest is located in the array. Two main approaches to search algorithm are the linear search approach and the binary search approach. Both methods have pros and cons in implementation, however, they are both successful in finding the index value of the point of interest in the given array.

The linear search algorithm checks the elements in the given array sequentially. This method can be used on both unsorted and sorted arrays, however, can be very time consuming. The binary search algorithm can be very quick in finding the point of interest index value, but the given array must be in ascending order. The binary search algorithm works but cutting the array in half and compare the point of interest to that midpoint. If the point of interest is greater than the midpoint then the new array is the upper half of the given array. Then the process is repeated until the midpoint value is equal to the point of interest, thus the index value for the point of interest is found.

The performance results of implementing the binary search and linear search methodologies against an array of size 10E8 are given in the Figure below.

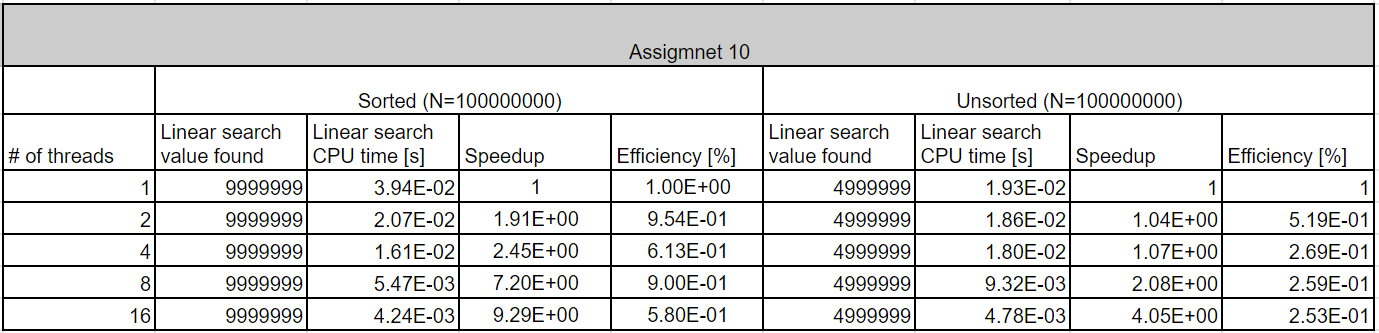


*Figure 1. Problem 9 Results*

For the sorted array, both search algorithms correctly identified the index value, however, the binary search algorithm was much faster in doing so, finding the value in 0.0003 seconds compared to the linear search method time of 0.04 seconds. For the unsorted array, the binary search method could not be implemented, but the linear search algorithm was and found the index value for the unsorted array successfully.

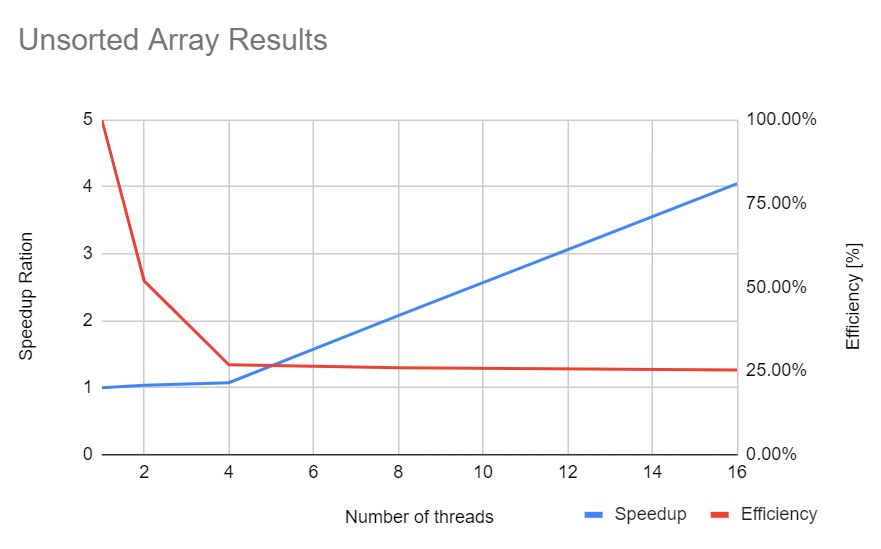
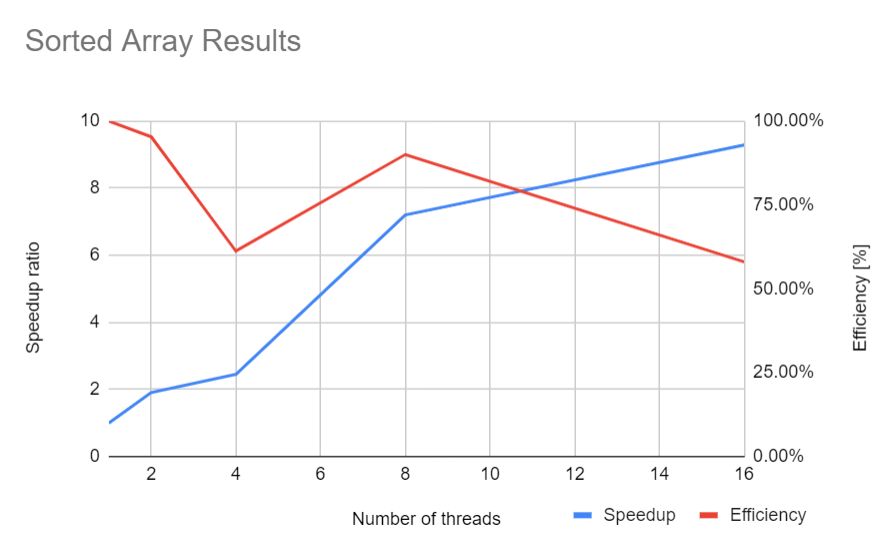
**Problem 2**

The upside to the linear search algorithm is that it is not limited by the input array type (sorted versus unsorted), but the downside is that it can be very slow. However, the compute time can be improved if the algorithm is parallelized. The parallelization of the linear search algorithm was implemented and tested against an input array of size 10E8 (for both sorted and unsorted arrays). The results of the parallelization for different number of threads are shown in the Figure below.



*Figure 2. Problem 10 Results*

As expected, the speedup time increased as the number of threads increased for this application for both types of input arrays. The speedup time and efficiency trends are depicted in the two figures below.

**

*Figure 3. Sorted Array Type Results Figure 4. Sorted Array Type Results*