Matthew Trembley

Southern New Hampshire University

10/10/2021

CS-260-R1998

6-1 Programming Journal

Binary tree algorithms seem to be very complicated, especially if one does not have a firm grasp on pointers. I do feel that they are very much worth the effort. The incredibly fast search time is very helpful when one needs to find a specific item within a list. “Divide and conquer” algorithms have 3 steps, or characteristics, that make them work. The first is to divide the larger problem into smaller, more palatable problems. Next, the algorithm will recursively solve these problems. And finally, the algorithm will attempt to recombine the solutions into one solution for the original problem. Now a binary search will take a list, and essentially mathematically equate each decision until it reaches the desired solution, resulting in much faster search results.

If each node of a tree is large, then available memory may be a limiting factor. To help with this, the trees could be broken up into smaller sections, in almost a blend of a divide and conquer algorithm and a binary search. If the large tree is broken up and each smaller tree is searched with a binary search, then it can then be put back together to obtain the correct result. This is where I feel one competency of this course comes into play – using different algorithms together to create new ones that can efficiently do a given task.