Kripash Shrestha

05/04/2017

CS 202 Spring 2017

Project 8 Documentation

The purpose of this project is to work with STL std::vectors and with recursive algorithms. The project will implement a quicksort algorithm, and a binary search algorithm to properly sort a vector of 100 integers and to find a target value within that sorted vector.

To quick sort the algorithm, I divided it into two parts as seen on the wiki page. A partition function and the actual sorting function. The partition function will set a target pivot, which will be the value of the last element in the entire vector. The function will then set i to the first element -1. The function will then go in a for loop and swaps the elements based on the logic, if the vector[j] (which j is used to for loop until the end) is smaller than the value, which was the value of the last element in the vector. The function will return i+1. The function is then topped off with the vector_resort function which checks if the integer corresponding to the first and last element of the vector. If the last element is greater than the first, then the function will call the partition function and assign the return type to a local variable. The function will then call itself with the parameters of the same vector, with the first element and then the local variable part -1. The function will call itself again with the parameters of the same vector, local variable part + 1 and with the last element. This is done recursively since the function will continuously call itself until the vector has been sorted properly.

The binary search part of the project was much easier in my opinion. Find the middle of the vector and if the target value is equal to that, then return that element. If not then check two conditions, the mid value is greater than the target value, then that means, we need to call the function but with parameters made to be from the first element to mid-1. Another condition is, if the mid value is smaller than the target value, which means, we need to call the function but with parameters made to be from the mid +1 to the last element. This is done recursively until the value is found. If the value is not found, which will be the case that left will be greater than right, then the function will return -1.