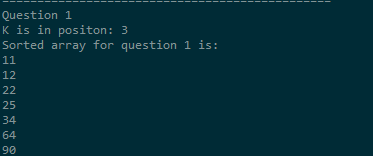
Lab 02 Report

The purpose of this lab was to implement different sorting algorithms and make a minor adjustment to each of them. The first algorithm we are asked to implement, is bubble sort that then returns the element in position k. The second question ask that we implement quick sort and also returns the element in position k. the third question ask that we implement a modified version of quicksort that only makes one recursive call using a pivot.

The way I went about implementing bubble sort in the first question was by doing it iteratively. First by traversing through all the elements in the list. then traverse the array from 0 to n-i-1 and then swap if the element found is greater than the next element. To return the position of k I created a separate for loop in the function to loop through the sorted list and when k is equal to the point in the list, return i. the run time for this function is O(n^2)

The next function we are asked to implement is quicksort. I went about this by having a low and a high variable. The first check is if low is less than high. I then initialize a separate variable ‘pi’ which is then set to the function partition which takes the last element as pivot, and places the pivot element at its correct position in sorted array and all the places smaller to the left of pivot and all greater elements to the right of pivot. Once the partition function finishes there are two recursive calls made where the first makes the pi variable high and -1 and same with the second but with the low variable. To return the position of k I loop the sorted array and check if the element is equal to k and return I if it is. Runtime is O(n^2)

The next function is quick\_modified\_sort which ask to implement quick sort using only one recursive call. In doing so I used the same quick sort function where low is compared to high and is low is less than high. If it is I set the ‘pi’ variable to the partition function again and does the minus one and plus one recursive call along with the pi variable in the high or low variable. I use an else statement to make the second call along with the same variable but this time named as p2 since they are separate calls. The runtime of this function is O(logn)

Screenshots for each output

