DSA – Practical: Quick Sort Evaluation

Refer to the notes on sorting algorithms to implement the Quick Sort and compare it with other sorting methods.

1. Write the implementation of the following functions:

```
// Sort array of integers in ascending order using quick sort algorithm
void quickSort( int data[], int size ){
       quickSortHelp(data, 0, size-1);
}

//Uses in-place partitioning to partition the array range from start to end
and recursively sort the resulting two partitions
void quickSortHelp(int data[], int start, int end){
    ...
}

//Tests if the given array is sorted or not. Returns 1 or 0.
int isSorted(int arr[], int size){
    ...
}
```

2. Use the isSorted() function to test whether all of your sorting functions are working correctly or not, by using it on the after-sort arrays.

3. Evaluate and compare the efficiencies of sorting methods you have implemented so far. Test sorting random arrays of varying sizes and record experiment results (in milliseconds) in the given table below, and discuss your findings:

Array size	bubbleSort	selectSort	quickSort
100			
1,000			
10,000			
…			

- The End -