

Bangladesh Army International University of Science & Technology
Department of Computer Science and Engineering

Lab Report

Lab Report No	02						
Lab Report Name	Sorting an Array Using Quick Sort						
Course Title	Computer Algorithms & Complexity Sessional						
Course Code	CSE 222						
Name	Rayhan Alam Towhid						
ID	1119007						
Level	2	Term	II	Section	A	Group	G1
Date of Submission	19-10-2025			Session	Fall-2025		

Marking Rubric:

Problem Understanding & Report Clarity (3)	Implementation (5)	Results & Analysis (2)	Total (10)

Key Learnings:

Quick Sort teaches how to divide and conquer by choosing a pivot, partitioning the array around it, and recursively sorting the subarrays. You also learn about in-place sorting and improving efficiency with recursion.

Code Implementation:



```
 1 #include <iostream>
 2 using namespace std;
 3
 4 void quickSort(int arr[], int low, int high)
 5 {
 6     if (low > high)
 7         return;
 8
 9     int pivot = arr[high];
10     int i = low;
11
12     for (int j = low; j < high; j++)
13     {
14         if (arr[j] < pivot)
15         {
16             int temp = arr[i];
17             arr[i] = arr[j];
18             arr[j] = temp;
19             i++;
20         }
21     }
22
23     int temp = arr[i];
24     arr[i] = arr[high];
25     arr[high] = temp;
26
27     quickSort(arr, low, i - 1);
28     quickSort(arr, i + 1, high);
29 }
30
31 int main()
32 {
33     int arr[] = {64, 34, 25, 12, 22, 11, 90, 5};
34     int size = sizeof(arr) / sizeof(arr[0]);
35
36     quickSort(arr, 0, size - 1);
37
38     for (int i = 0; i < size; i++)
39     {
40         cout << arr[i] << " ";
41     }
42     cout << endl;
43
44     return 0;
45 }
```

Sample Input - Output:

```
PS F:\All Codes\JavaScript> cd "f:\All Codes"
• 5 11 12 22 25 34 64 90
PS F:\All Codes\JavaScript>
```

Result Analysis / Discussion:

Quick Sort sorts an array by selecting a pivot element and placing smaller elements before it and larger elements after it. This process is repeated recursively for the subarrays. It is efficient, works in-place, and has an average time complexity of $O(n \log n)$.