

Apex College

BCIS Program

Affiliated to Pokhara University



Data Structure & Algorithms

Lab Report

14

Quick Sorting

Date: 05-07-2022

checked
5/7/22
8.

Submitted by:

Ishwor Shrestha

Roll no.: 2018-BCIS-414

Submitted to:

Pravakar Ghimire, &

Anmol Shrestha

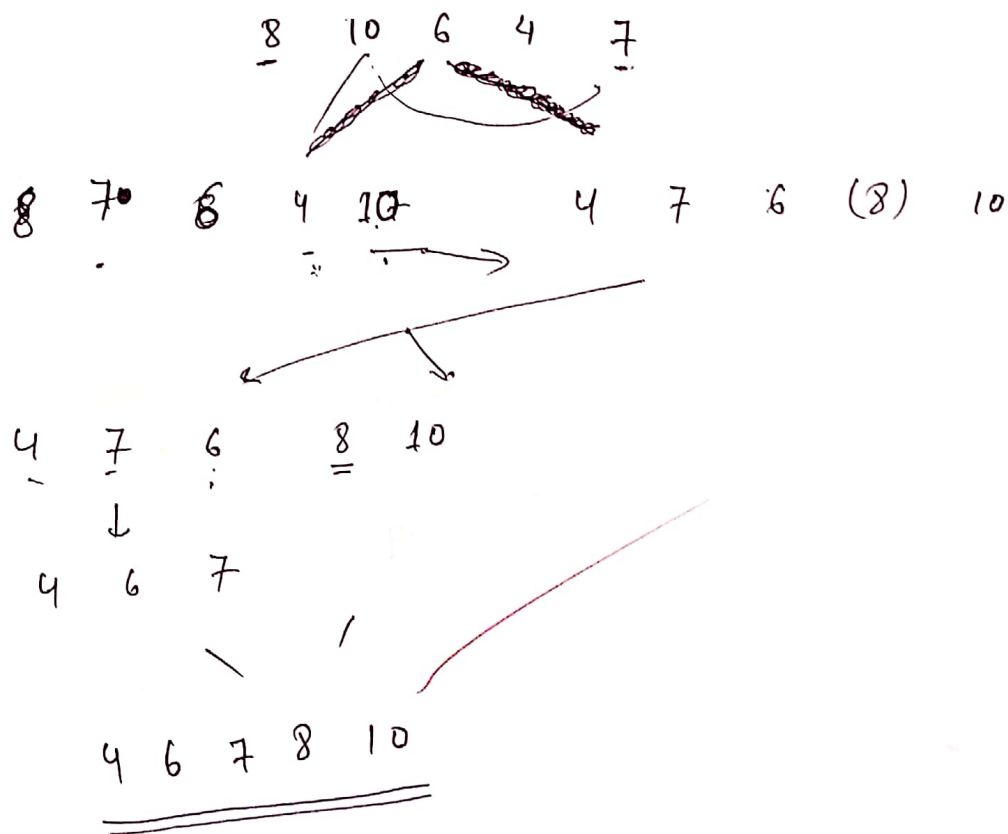
Apex College

#Lab 14 Objectives

- To implement quick sort algorithm to sort list of data.

#Introduction

Quick Sort is a in-place and ~~not~~unstable sorting algorithm to sort data in a list. It is a divide and conquer algorithm which picks an element as pivot and partitions the given array around picked pivot.



Source Code

```
#include <stdio.h>
#include <stdlib.h>

int r, pivot;
```

```

void swap (int arr[], int x, int y) {
    int temp;
    temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}

```

```

void printArray (int arr[], int size) {
    for (i=0; i<size; i++)
        printf ("%d", arr[i]);

    printf ("\n");
}

```

```

int partition (int arr[], int l, int r) {
    int x, y;
    x = l;
    y = r;
    pivot = arr[l];
    while (x < y) {
        while (arr[x] <= pivot)
            x++;

        while (arr[y] > pivot)
            y--;

        if (x < y)
            swap (arr, x, y);
    }
    arr[l] = arr[y];
    arr[y] = pivot;
    return y;
}

```

```

void quickSort (int arr[], int l, int r) {
    if (l < r) {
        pivot = partition(arr, l, r);
        quickSort(arr, l, pivot-1);
        quickSort(arr, pivot+1, r);
    }
}

```

```

int main() {
    int l = 0;
    // printf("Enter numbers of Elements");

    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);
    int r = n-1;

    quickSort(arr, l, r);
    printf("Sorted array: \n");
    printArray(arr, n);

    return 0;
}

```

Activities

We performed quick sort using partition & swap.

Conclusion

I learned about quick sorting algorithm using partition & swap technique.