Apex College

BCIS Program

Affiliated to Pokhara University

Data Structure & Algorithms
Lab Report
13

Bubble, Insertion, Selection Sort Algorithms

Date: 01-07-2022

1 Late checked 20/7/22 S.

Submitted by:
Ishwor Shrestha
Roll no.: 2018-BCIS-414

Submitted to:
Pravakar Ghimire, &
Anmol Shrestha
Apex College

#Lab13 Objectives

- To implement the bubble, insertion and selection sorting algorithms.

#Introduction

Sorting is an algorithm that puts elements of a list into an order. Sorting follows mainly two properties i.e. Inplace vs non-inplace and stable vs. non stable.

- Bubble sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.

  Since bubble sort algorithm have $O(n^2)$ of time complexity it is one of the worst sorting algorithm.

- Insertion sort is a sorting algorithm in which the elements are transferred one at a time to the right position.

  If the given list is already sorted, then the time complexity is $O(n)$ otherwise $O(n^2)$

- Selection sort is a sorting algorithm which works on the idea of repeatedly finding the smallest element and placing it at its correct sorted position.

  Time complexity : $O(n^2)$

Source Code

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

void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y = temp;
}
void printArray(int arr[], int size){
    int i;
    for(i=0; i<size; i++)
        printf("%d", arr[i]);
    printf("\n");
}

void bubbleSort(int arr[], int n){
    int i, j;
    for(i=0; i<n-1; i++)
        for(j=0; j<n-i-1; j++)
            if(arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}
void insertionSort(int arr[], int n){
    int i, j;
    for(i=0; i<=n-1; i++)
        for(j=i-1; j>=0; j--)
            if(arr[j-1] > arr[j])
                swap(&arr[j-1], &arr[j]);
}
```

```
void selectionSort(int arr[], int n) {
    int i, j;
    for (i=0; i<n-1; i++) {
        int least = arr[i];
        int p = j;
        for (j=i+1; j<n; j++) {
            if (arr[j] < least) {
                least = arr[j];
            }
            p=j;
        }
        swap(&arr[i], &arr[p]);
    }
}

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr) / sizeof(arr[0]);
    int ch;
    while (1) {
        printf("1. Bubble Sort \n 2. Insertion Sort \n 3.
                selection sort \n 4. Exit \n");
        printf("Enter your choice for Sorting: ");
        scanf("%d", &ch);

        switch(ch) {
            case 1:
                bubbleSort(arr, n);
```

```
        printf("Sorted array using Bubble sort:\n");
        printArray(arr, n);
        break;

    case 2:
        InsertionSort(arr, n);
        printf("Sorted array using Insertion sort:\n");
        printArray(arr, n);
        break;

    case 3:
        SelectionSort(arr, n);
        printf("Sorted array using Selection sort:\n");
        printArray(arr, n);
        break;

    case 4:
        exit(0);

    default:
        printf("Invalid option\n");
    }

}

return 0;

}
```

#Activities:
We performed 3 different sorting algorithms to sort an array list.

#Conclusion:
I learned about the bubble, selection & insertion sort