

Name: ZISHNENDU SARKER

Roll: 2K19/CO/450

Java Programming

Lab assignment 09

28/11/2021

Java Program to Implement Bubble Sort algorithm

Theory: Java program to implement bubble sort Java Programming Java8 Java Technologies
Bubble sort is a simple sorting algorithm. This sorting algorithm is a comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.

Code:

```
// import the Class
import java.util.Arrays;
import java.util.Scanner;

class Main {

    // create an object of scanner
    // to take input from the user
    Scanner input = new Scanner(System.in);

    // method to perform bubble sort
    void bubbleSort(int array[]) {
        int size = array.length;

        // for ascending or descending sort
        System.out.println("Choose Sorting Order:");
        System.out.println("1 for Ascending \n2 for Descending");
        int sortOrder = input.nextInt();

        // run loops two times
        // first loop access each element of the array
        for (int i = 0; i < size - 1; i++)
```

```

// second loop performs the comparison in each iteration
for (int j = 0; j < size - i - 1; j++)

    // sort the array in ascending order
    if (sortOrder == 1) {
        // compares the adjacent element
        if (array[j] > array[j + 1]) {

            // swap if left element is greater than right
            int temp = array[j];
            array[j] = array[j + 1];
            array[j + 1] = temp;
        }
    }

    // sort the array in descending order
    else {
        // compares the adjacent element
        if (array[j] < array[j + 1]) {

            // swap if left element is smaller than right
            int temp = array[j];
            array[j] = array[j + 1];
            array[j + 1] = temp;
        }
    }
}

// driver code
public static void main(String args[]) {

    // create an array
    int[] data = { -2, 45, 0, 11, -9 };

    // create an object of Main class
    Main bs = new Main();

    // call the method bubbleSort using object bs
    // pass the array as the method argument

```

```
bs.bubbleSort(data);
System.out.println("Sorted Array in Ascending Order:");

// call toString() of Arrays class
// to convert data into the string
System.out.println(Arrays.toString(data));
}
}
```

Output:

```
Choose Sorting Order:
1 for Ascending
2 for Descending
1
Sorted Array:
[-9, -2, 0, 11, 45]
```

Java Program to Implement Quick Sort Algorithm

Theory: Quick Sort algorithm can be implemented in Java by forming a pseudo code with a sequence of steps designed and followed in an efficient manner. The main principle of the quick sort algorithm that it works is based on the divide and conquer approach and is also an efficient sorting algorithm.

Code:

```
import java.util.Arrays;

class Quicksort {

    // method to find the partition position
    static int partition(int array[], int low, int high) {

        // choose the rightmost element as pivot
        int pivot = array[high];
```

```

// pointer for greater element
int i = (low - 1);

// traverse through all elements
// compare each element with pivot
for (int j = low; j < high; j++) {
    if (array[j] <= pivot) {

        // if element smaller than pivot is found
        // swap it with the greater element pointed by i
        i++;

        // swapping element at i with element at j
        int temp = array[i];
        array[i] = array[j];
        array[j] = temp;
    }
}

// swap the pivot element with the greater element specified by i
int temp = array[i + 1];
array[i + 1] = array[high];
array[high] = temp;

// return the position from where partition is done
return (i + 1);
}

static void quickSort(int array[], int low, int high) {
    if (low < high) {

        // find pivot element such that
        // elements smaller than pivot are on the left
        // elements greater than pivot are on the right
        int pi = partition(array, low, high);

        // recursive call on the left of pivot
        quickSort(array, low, pi - 1);
    }
}

```

```

        // recursive call on the right of pivot
        quickSort(array, pi + 1, high);
    }
}
}

// Main class
class Main {
    public static void main(String args[]) {

        int[] data = { 8, 7, 2, 1, 0, 9, 6 };
        System.out.println("Unsorted Array");
        System.out.println(Arrays.toString(data));

        int size = data.length;

        // call quicksort() on array data
        Quicksort.quickSort(data, 0, size - 1);

        System.out.println("Sorted Array in Ascending Order ");
        System.out.println(Arrays.toString(data));
    }
}

```

Output:

```

Unsorted Array
[8, 7, 2, 1, 0, 9, 6]
Sorted Array in Ascending Order
[0, 1, 2, 6, 7, 8, 9]

```

Java Program to Implement Binary Search Algorithm

Theory: Java search algorithm programs Program: Implement Binary search in java using recursive algorithm. A binary search or half-interval search algorithm finds the position of a specified value (the input "key") within a sorted array. In each step, the algorithm compares the input key-value with the key value of the middle element of the array.

Code:

```
public class BinarySearch {  
    public static void main(String args[]){  
        int array[] = {3, 20, 25, 57, 63, 96};  
        int size = array.length;  
        int low = 0;  
        int high = size-1;  
        int value = 25;  
        int mid = 0;  
        mid = low +(high-low)/2;  
  
        while(low<=high){  
            if(array[mid] == value){  
                System.out.println(mid);  
                break;  
            }  
            else if(array[mid]<value)  
                low = mid+1;  
            else high = mid - 1;  
        }  
        mid = (low+high)/2;  
    }  
}
```

Output:

```
Enter element to be searched:  
6  
Element found at index 3
```