**Day-9 Basics of Java**

Sorting & Searching Algorithms in Java

1. **package** com.mycom.sorting.quicksort;

**import** java.util.Scanner;

**public** **class** QuickSort {

**public** **static** **void** quickSort(**int**[] arr, **int** low, **int** high) {

**if** (low < high) {

**int** pi = *partition*(arr, low, high);

*quickSort*(arr, low, pi - 1);

*quickSort*(arr, pi + 1, high);

}

}

**public** **static** **int** partition(**int**[] arr, **int** low, **int** high) {

**int** pivot = arr[high];

**int** i = (low - 1);

**for** (**int** j = low; j < high; j++) {

**if** (arr[j] < pivot) {

i++;

**int** temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

**int** temp = arr[i + 1];

arr[i + 1] = arr[high];

arr[high] = temp;

**return** i + 1;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter no of elements: ");

**int** n = sc.nextInt();

**int**[] arr = **new** **int**[n];

System.***out***.println("Enter array elements:");

**for** (**int** i = 0; i < n; i++) {

arr[i] = sc.nextInt();

}

*quickSort*(arr, 0, arr.length - 1);

System.***out***.println("Sorted array elements:");

**for** (**int** num : arr) {

System.***out***.print(num + " ");

}

sc.close();

}

}

2. **package** com.mycom.sorting.mergesort;

**import** java.util.Scanner;

**public** **class** MergeSort {

**public** **static** **void** merge(**int**[] arr, **int** left, **int** middle, **int** right) {

**int** n1 = middle - left + 1;

**int** n2 = right - middle;

**int**[] L = **new** **int**[n1];

**int**[] R = **new** **int**[n2];

**for** (**int** i = 0; i < n1; ++i)

L[i] = arr[left + i];

**for** (**int** j = 0; j < n2; ++j)

R[j] = arr[middle + 1 + j];

**int** i = 0, j = 0;

**int** k = left;

**while** (i < n1 && j < n2) {

**if** (L[i] <= R[j]) {

arr[k] = L[i];

i++;

} **else** {

arr[k] = R[j];

j++;

}

k++;

}

**while** (i < n1) {

arr[k] = L[i];

i++;

k++;

}

**while** (j < n2) {

arr[k] = R[j];

j++;

k++;

}

}

**public** **static** **void** sort(**int**[] arr, **int** left, **int** right) {

**if** (left < right) {

**int** middle = (left + right) / 2;

*sort*(arr, left, middle);

*sort*(arr, middle + 1, right);

*merge*(arr, left, middle, right);

}

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.println("Enter no of array elements:");

**int** n = scanner.nextInt();

**int**[] arr = **new** **int**[n];

System.***out***.println("Enter array elements:");

**for** (**int** i = 0; i < n; i++) {

arr[i] = scanner.nextInt();

}

*sort*(arr, 0, arr.length - 1);

System.***out***.println("Sorted array elements:");

**for** (**int** num : arr) {

System.***out***.print(num + " ");

}

scanner.close();

}

}

3. **package** com.mycom.sorting.linearsearch;

**import** java.util.Scanner;

**public** **class** LinearSearch {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of elements in the array: ");

**int** n = scanner.nextInt();

**int**[] arr = **new** **int**[n];

System.***out***.print("Enter the elements of the array: ");

**for** (**int** i = 0; i < n; i++) {

arr[i] = scanner.nextInt();

}

System.***out***.print("Enter the element to search for: ");

**int** x = scanner.nextInt();

**boolean** found = **false**;

**for** (**int** i = 0; i < n; i++) {

**if** (arr[i] == x) {

found = **true**;

**break**;

}

}

**if** (found) {

System.***out***.println("Element is Present");

} **else** {

System.***out***.println("Element is not Present");

}

scanner.close();

}

}

**4.package** com.mycom.sorting.selectionsort;

**public** **class** SelectionSortStrings{

**public** **static** **void** main(String[] args) {

String[] fruits = {"banana", "apple", "orange", "grape", "kiwi"};

*selectionSort*(fruits);

**for**(String fruit:fruits ) {

   System.***out***.print(fruit+" ");

}

System.***out***.println("\n");

**for** (String fruit : fruits) {

System.***out***.print(fruit + " ");

}

}

**public** **static** **void** selectionSort(String[] arr) {

**int** n = arr.length;

**for** (**int** i = 0; i < n - 1; i++) {

**int** minIndex = i;

**for** (**int** j = i + 1; j < n; j++) {

**if** (arr[j].compareTo(arr[minIndex]) < 0) {

minIndex = j;

}

}

String temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

}

}

}

**5. package** com.mycom.sorting.Insertionsort;

**import** java.util.Scanner;

**public** **class** InsertionSortFloats {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of floating-point numbers: ");

**int** n = scanner.nextInt();

**double**[] arr = **new** **double**[n];

System.***out***.println("Enter the floating-point numbers: ");

**for** (**int** i = 0; i < n; i++) {

arr[i] = scanner.nextDouble();

}

*insertionSort*(arr);

System.***out***.println("Sorted array: ");

**for** (**double** num : arr) {

System.***out***.print(num + " ");

}

scanner.close();

}

**public** **static** **void** insertionSort(**double**[] arr) {

**int** n = arr.length;

**for** (**int** i = 1; i < n; ++i) {

**double** key = arr[i];

**int** j = i - 1;

**while** (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

}

6. **package** com.mycom.sorting.string;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** BinarySearchString {

**public** **static** **void** main(String[] args) {

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Enter the strings (space-separated): ");

String input = scanner.nextLine();

String[] arr = input.split(" ");

System.***out***.print("Enter the string to search for: ");

String target = scanner.nextLine();

Arrays.*sort*(arr);

**int** result = *binarySearch*(arr, target);

**if** (result == -1) {

System.***out***.println("Element " + target + " not found");

} **else** {

System.***out***.println("Element " + target + " found at index " + result);

}

scanner.close();

}

**public** **static** **int** binarySearch(String[] arr, String target) {

**int** left = 0, right = arr.length - 1;

**while** (left <= right) {

**int** mid = left + (right - left) / 2;

**int** res = target.compareTo(arr[mid]);

**if** (res == 0) {

**return** mid;

}

**if** (res > 0) {

left = mid + 1;

}

**else** {

right = mid - 1;

}

}

**return** -1;

}

}