**Sorting & Searching Algorithms in Java**

1. **Write a Program to sort an array of given integers using the Quick sort.**

package day9assignment;

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) {

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

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

int n = scanner.nextInt();

int[] arr = new int[n];

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

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

arr[i] = scanner.nextInt();

}

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

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

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

System.*out*.print(arr[i] + " ");

}

}

}

**Output:**

Enter the number of elements: 6

Enter the elements:

10 7 8 9 1 5

Sorted array:

1 5 7 8 9 10

1. **Write a Program to sort an array of elements using Merge sort.**

package day9assignment;

import java.util.Scanner;

public class Mergesort {

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

if (left < right) {

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

*mergeSort*(arr, left, mid);

*mergeSort*(arr, mid + 1, right);

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

}

}

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

int n1 = mid - left + 1;

int n2 = right - mid;

int[] leftArray = new int[n1];

int[] rightArray = new int[n2];

System.*arraycopy*(arr, left, leftArray, 0, n1);

System.*arraycopy*(arr, mid + 1, rightArray, 0, n2);

int i = 0, j = 0, k = left;

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

if (leftArray[i] <= rightArray[j]) {

arr[k] = leftArray[i];

i++;

} else {

arr[k] = rightArray[j];

j++;

}

k++;

}

while (i < n1) {

arr[k] = leftArray[i];

i++;

k++;

}

while (j < n2) {

arr[k] = rightArray[j];

j++;

k++;

}

}

public static void main(String[] args) {

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

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

int n = scanner.nextInt();

int[] arr = new int[n];

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

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

arr[i] = scanner.nextInt();

}

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

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

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

System.*out*.print(arr[i] + " ");

}

}

}

**Output:**

Enter the number of elements: 6

Enter the elements:

11 12 13 5 6 7

Sorted array:

5 6 7 11 12 13

1. **Write a Program to find a specific element in an array of elements using Linear Search.**

package day9assignment;

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: ");

int n = scanner.nextInt();

int[] arr = new int[n];

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

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

arr[i] = scanner.nextInt();

}

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

int num = scanner.nextInt();

int index = *linearSearch*(arr, num);

if (index == -1) {

System.*out*.println("Element not present.");

} else {

System.*out*.println("Element present. ");

}

}

public static int linearSearch(int[] arr, int num) {

for (int i = 0; i < arr.length; i++) {

if (arr[i] == num) {

return i;

}

}

return -1;

}

}

**Output:**

Enter the number of elements: 5

Enter the elements:

2 3 4 10 40

Enter the element to search: 10

Element present.

**Output:**

Enter the number of elements: 5

Enter the elements:

2 3 4 10 40

Enter the element to search: 50

Element not present.

1. **Write a Program to sort an array of strings using selection sort.**

**package** day9assignment;

**import** java.util.Scanner;

**public** **class** Selectionsort {

**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[i];

arr[i] = arr[minIndex];

arr[minIndex] = temp;

}

}

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

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

System.***out***.print("Enter the number of strings: ");

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

scanner.nextLine();

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

System.***out***.println("Enter the strings:");

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

arr[i] = scanner.nextLine();

}

*selectionSort*(arr);

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

**for** (String str : arr) {

System.***out***.println(str);

}

}

}

**Output:**

Enter the number of strings: 5

Enter the strings:

banana

apple

orange

mango

kiwi

Sorted strings:

apple

banana

kiwi

mango

orange

1. **Write a Program to sort an array of floating-point numbers using insertion sort.**

**package** day9assignment;

**import** java.util.Scanner;

**public** **class** Insertionsort {

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

**int** n = arr.length;

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

**float** 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;

}

}

**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();

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

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

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

arr[i] = scanner.nextFloat();

}

*insertionSort*(arr);

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

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

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

}

}

}

**Output:**

Enter the number of floating-point numbers: 5

Enter the floating-point numbers:

12.5 11.0 13.2 5.7 6.1

Sorted floating-point numbers:

* 1. 6.1 11.0 12.5 13.2

1. **Implement binary search to find a specified string within a sorted array of strings.**

package day9assignment;

import java.util.Scanner;

public class Binarysearch {

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

int left = 0;

int right = arr.length - 1;

while (left <= right) {

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

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

if (comparison == 0) {

return mid;

}

if (comparison < 0) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return -1;

}

public static void main(String[] args) {

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

System.*out*.print("Enter the number of strings: ");

int n = scanner.nextInt();

scanner.nextLine();

String[] arr = new String[n];

System.*out*.println("Enter the strings in sorted order:");

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

arr[i] = scanner.nextLine();

}

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

String target = scanner.nextLine();

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

if (result == -1) {

System.*out*.println("String not found.");

} else {

System.*out*.println("String found at index: " + result);

}

}

}

**Output:**

Enter the number of strings: 5

Enter the strings in sorted order:

apple

banana

mango

orange

kiwi

Enter the string to search: mango

String found at index: 2

**Output:**

Enter the number of strings: 5

Enter the strings in sorted order:

apple

banana

mango

orange

kiwi

Enter the string to search: pineapple

String not found.