Task 6

a)

import java.util.\*;

public class CountingSort {

public static void countingSort(int[] arr) {

int max = Arrays.stream(arr).max().getAsInt();

int min = Arrays.stream(arr).min().getAsInt();

int range = max - min + 1;

int[] count = new int[range];

int[] output = new int[arr.length];

for (int num : arr) count[num - min]++;

for (int i = 1; i < count.length; i++) {

count[i] += count[i - 1];

}

for (int i = arr.length - 1; i >= 0; i--) {

output[count[arr[i] - min] - 1] = arr[i];

count[arr[i] - min]--;

}

System.arraycopy(output, 0, arr, 0, arr.length);

}

public static void main(String[] args) {

int[] arr = {4, 2, 2, 8, 3, 3, 1};

countingSort(arr);

System.out.println("Sorted: " + Arrays.toString(arr));

}

}

b)

import java.util.\*;

public class RadixSort {

public static void countingSort(int[] arr, int exp) {

int n = arr.length;

int[] output = new int[n];

int[] count = new int[10];

for (int num : arr) count[(num / exp) % 10]++;

for (int i = 1; i < 10; i++) count[i] += count[i - 1];

for (int i = n - 1; i >= 0; i--) {

int idx = (arr[i] / exp) % 10;

output[count[idx] - 1] = arr[i];

count[idx]--;

}

System.arraycopy(output, 0, arr, 0, n);

}

public static void radixSort(int[] arr) {

int max = Arrays.stream(arr).max().getAsInt();

for (int exp = 1; max / exp > 0; exp \*= 10) {

countingSort(arr, exp);

}

}

public static void main(String[] args) {

int[] arr = {170, 45, 75, 90, 802, 24, 2, 66};

radixSort(arr);

System.out.println("Sorted: " + Arrays.toString(arr));

}

}

c)

import java.util.\*;

public class HeapSort {

public static void heapify(int[] arr, int n, int i) {

int largest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && arr[left] > arr[largest]) largest = left;

if (right < n && arr[right] > arr[largest]) largest = right;

if (largest != i) {

int temp = arr[i];

arr[i] = arr[largest];

arr[largest] = temp;

heapify(arr, n, largest);

}

}

public static void heapSort(int[] arr) {

int n = arr.length;

// Build max heap

for (int i = n / 2 - 1; i >= 0; i--) {

heapify(arr, n, i);

}

// Extract elements

for (int i = n - 1; i > 0; i--) {

int temp = arr[0];

arr[0] = arr[i];

arr[i] = temp;

heapify(arr, i, 0);

}

}

public static void main(String[] args) {

int[] arr = {12, 11, 13, 5, 6, 7};

heapSort(arr);

System.out.println("Sorted: " + Arrays.toString(arr));

}

}