Problems based on sortings

Assignment Solutions

PW SKILLS

Q1 - Given an array of strings arr[]. Sort given strings using Bubble Sort and display the sorted array.

Input:

n = 3

arr[] = [coding, is, fun]

Expected Output:

coding fun is

Explanation:

- Traverse through the array and use bubble sort algorithm.
- To compare strings lexicographically, we use .compareTo() method of java.

Code:

https://pastebin.com/495fXqGr

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the length of array
3
Enter the elements of array
coding is fun
coding fun is
Process finished with exit code 0
```

Q2- Given a string s, return the string which contains all characters of s in lexicographical order.

Input:

coding

Expected Output:

cdgino

Explanation:

- Convert string to a character array, sort the array then return it in a string format.

Code:

https://pastebin.com/8d69xGx2

```
/Library/Java/JavaVirtualMachines/jdk-19.jc
Enter the string
coding
cdgino
Process finished with exit code 0
```

Q3 - Given an integer array and an integer k where k<size of array, We need to return the kth smallest element of the array.

Input1:

n = 5

arr[] = [3,5,6,2,1]

k=3

Expected Output:

3

Explanation:

- Sort the array using any sorting technique, we have used insertion sort.
- Print element at k-1 index in the sorted array.

Code:

https://pastebin.com/n41ASZKv

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk/C
Enter the length of array
5
Enter the elements of array
3 5 6 2 1
Enter the index
3
3
Process finished with exit code 0
```

Q4 - Given an array of n elements, the task is to find the elements that are greater than half of elements in an array. In case of odd elements, we need to print elements larger than floor(n/2) elements where n is total number of elements in array.

Input:

n = 4

arr[] = [1, 6, 3, 4]

Expected Output:

4 6

Explanation:

- Sort the array in ascending order and then print last ceil(n/2) elements from the sorted array.
- Suppose if n is odd, then according to the question, we want elements that are greater than n/2 elements (here since n is odd then n/2 will be the floor value).
- Let's take an example to understand this:
- Suppose the value of n = 7. Now we want those numbers whose value is greater than floor(7/2) i.e. 3. Therefore elements remaining as answer would be 7 - floor(7/2) = 7 - 3 = 4.
- Now 4 is nothing but ceil(7/2).
- Therefore it was earlier mentioned that we need to consider ceil(n/2).
- In the case of even numbers of elements there is no difference in floor and ceil value. The only difference arises because of odd numbers of elements and that is already sorted.

Code:

https://pastebin.com/7mEc5ET8

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the length of array
4
Enter the elements of array
1 6 3 4
4 6
Process finished with exit code 0
```

Q5 - Given an array of digits (values are from 0 to 9), the task is to find the minimum possible sum of two numbers formed from digits of the array. Please note that all digits of the given array must be used to form the two numbers.

Input:

n = 6

arr[] = [6, 8, 4, 5, 2, 3]

Expected Output:

604

Explanation:

- Sort the array in ascending order
- Create two number by picking a digit from sorted array alternatively i.e. from even and odd index

Code:

https://pastebin.com/TVK53gwH

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the length of the array
6
Enter the elements of the array
6 8 4 5 2 3
604

Process finished with exit code 0
```

Q6 - Given an array with N distinct elements, convert the given array to a form where all elements are in the range from 0 to N-1. The order of elements is the same, i.e., 0 is placed in the place of the smallest element, 1 is placed for the second smallest element, ... N-1 is placed for the largest element.

Input:

n = 3

arr[] = {10, 40, 20}

Expected Output:

0 2 1

Explanation:

- Create a cloned array temp[].
- Sort temp[] in ascending order.
- Traverse temp[] from left to right and search for the number in the original array.
- Replace this index with its position.

Code:

https://pastebin.com/uAybKswC

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the length of the array
3
Enter the elements of the array
10 40 20
0 2 1
Process finished with exit code 0
```

Q7- Given an array, sort it using insertion sort recursively.

Input:
n = 5
arr[ ] = [4,5,2,7,1]
Expected Output:
1 2 4 5 7
Explanation:

- Base Case: If array size is 1 or smaller, return.
- Recursively sort first n-1 elements.
- Insert last element at its correct position in sorted array.

Code:
https://pastebin.com/3Yphq3Rs

```
/Library/Java/JavaVirtualMachines/jdk-19.jdk
Enter the length of the array
5
Enter the elements of the array
4 5 2 7 1
1 2 4 5 7
Process finished with exit code 0
```