Assignment-64

Sorting Algorithms

1. Write a C++ Program to sort a linked list 0s, 1s or 2s.
2. Given an integer array and a positive integer k, count all distinct pairs with differences equal to k.

   Example 1:

   Input: arr[] = {1, 5, 3, 4, 2}, k = 3
   Output: 2

   There are 2 pairs with difference 3, the pairs are {1, 4} and {5, 2}

   Input: arr[] = {8, 12, 16, 4, 0, 20}, k = 4
   Output: 5

   There are 5 pairs with difference 4, the pairs are {0, 4}, {4, 8}, {8, 12}, {12, 16} and {16, 20}

3. Given a sorted array and a number x, find a pair in an array whose sum is closest to x.

   Example 1:

   Input: arr[] = {10, 22, 28, 29, 30, 40}, x = 54
   Output: 22 and 30

   Input: arr[] = {1, 3, 4, 7, 10}, x = 15
   Output: 4 and 10

4. Given an unsorted array of integers, sort the array into a wave array. An array arr[0..n-1] is sorted in wave form if:
   arr[0] >= arr[1] <= arr[2] >= arr[3] <= arr[4] >= …..

   Example :

   Input: arr[] = {10, 5, 6, 3, 2, 20, 100, 80}
   Output: arr[] = {10, 5, 6, 2, 20, 3, 100, 80}
   Explanation:
   Here you can see {10, 5, 6, 2, 20, 3, 100, 80} the first element is larger than the second and the same thing is repeated again and again. large element – small element-large element -small element and so on .it can be small element-larger element – small element-large element -small element too. all you need to maintain is the up-down fashion which represents a wave. there can be multiple answers.

5. An interval is represented as a combination of start time and end time. Given a set of intervals, check if any two intervals intersect.

Examples :

Input: arr[] = {{1, 3}, {5, 7}, {2, 4}, {6, 8}}
Output: true
The intervals {1, 3} and {2, 4} overlap

Input: arr[] = {{1, 3}, {7, 9}, {4, 6}, {10, 13}}
Output: false
No pair of intervals overlap.

6. Given an almost sorted array where only two elements are swapped, how to sort the array efficiently?

Examples :

Input: arr[] = {10, 20, 60, 40, 50, 30}
// 30 and 60 are swapped
Output: arr[] = {10, 20, 30, 40, 50, 60}

Input: arr[] = {10, 20, 40, 30, 50, 60}
// 30 and 40 are swapped
Output: arr[] = {10, 20, 30, 40, 50, 60}

7. Given an array of dates, how to sort them.

Examples :

Input:
Date arr[] = {{20, 1, 2014},
{25, 3, 2010},
{ 3, 12, 1676},
{18, 11, 1982},
{19, 4, 2015},
{ 9, 7, 2015}}

Output:
Date arr[] = {{ 3, 12, 1676},
{18, 11, 1982},
{25, 3, 2010},
{20, 1, 2014},
{19, 4, 2015},

{ 9, 7, 2015}}

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

9. Given an array, arr[0..n-1] of distinct elements and a range [low, high], find all numbers that are in a range, but not the array. The missing elements should be printed in sorted order.

Examples :

Input: arr[] = {10, 12, 11, 15},
low = 10, high = 15
Output: 13, 14

Input: arr[] = {1, 14, 11, 51, 15},
low = 50, high = 55
Output: 50, 52, 53, 54 55

10. 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.

Examples :

Input: arr[] = {10, 40, 20}
Output: arr[] = {0, 2, 1}
Input: arr[] = {5, 10, 40, 30, 20}
Output: arr[] = {0, 1, 4, 3, 2}