

School of Computer Science and Engineering

Experiment List for Programming Ability and Logic Building - 1

Proposed Date	Lecture	Experiment	In Class / Take Home
Week: (26/01/26 to 31/01/26)	1	<p>Given an integer array <code>arr[]</code> and an integer <code>k</code>, your task is to find and return the <code>k</code>th smallest element in the given array.</p> <p>Note: The <code>k</code>th smallest element is determined based on the sorted order of the array.</p> <p>Examples:</p> <p>Input: <code>arr[] = [10, 5, 4, 3, 48, 6, 2, 33, 53, 10]</code>, <code>k = 4</code> Output: 5 Explanation: 4th smallest element in the given array is 5.</p> <p>Input: <code>arr[] = [7, 10, 4, 3, 20, 15]</code>, <code>k = 3</code> Output: 7 Explanation: 3rd smallest element in the given array is 7.</p> <p>Constraints:</p> $1 \leq \text{arr.size()} \leq 10^5$ $1 \leq \text{arr}[i] \leq 10^5$ $1 \leq k \leq \text{arr.size()}$ <p>Link: https://www.geeksforgeeks.org/problems/kth-smallest-element5635/1</p>	In Class
	1	<p>Given an array <code>arr[]</code> denoting heights of <code>n</code> towers and a positive integer <code>k</code>.</p> <p>For each tower, you must perform exactly one of the following operations exactly once.</p> <p>Increase the height of the tower by <code>k</code> Decrease the height of the tower by <code>k</code> Find out the minimum possible difference between the height of the shortest and tallest towers after you have modified each tower.</p> <p>You can find a slight modification of the problem here.</p> <p>Note: It is compulsory to increase or decrease the height by <code>k</code> for each tower. After the operation, the resultant array should not contain any negative integers.</p> <p>Examples :</p> <p>Input: <code>k = 2</code>, <code>arr[] = [1, 5, 8, 10]</code> Output: 5 Explanation: The array can be modified as <code>[1+k, 5-k, 8-k, 10-k] = [3, 3, 6, 8]</code>. The difference between the largest and the smallest is <code>8-3 = 5</code>.</p> <p>Input: <code>k = 3</code>, <code>arr[] = [3, 9, 12, 16, 20]</code> Output: 11 Explanation: The array can be modified as <code>[3+k, 9+k, 12-k, 16-k, 20-k] = [6, 12, 9, 13, 17]</code>. The difference between the largest and the smallest is <code>17-6 = 11</code>.</p> <p>Constraints</p> $1 \leq k \leq 10^7$ $1 \leq n \leq 10^5$ $1 \leq \text{arr}[i] \leq 10^7$ <p>Link: https://www.geeksforgeeks.org/problems/minimize-the-heights3351/1</p>	In Class
	1	<p>You are given an array <code>arr[]</code> of non-negative numbers. Each number tells you the maximum number of steps you can jump forward from that position.</p> <p>For example:</p> <p>If <code>arr[i] = 3</code>, you can jump to index <code>i + 1</code>, <code>i + 2</code>, or <code>i + 3</code> from position <code>i</code>. If <code>arr[i] = 0</code>, you cannot jump forward from that position.</p> <p>Your task is to find the minimum number of jumps needed to move from the first position in the array to the last position.</p>	Take Home

School of Computer Science and Engineering

Experiment List for Programming Ability and Logic Building - 1

Proposed Date	Lecture	Experiment	In Class / Take Home
		<p>Note: Return -1 if you can't reach the end of the array.</p> <p>Examples :</p> <p>Input: arr[] = [1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9] Output: 3 Explanation: First jump from 1st element to 2nd element with value 3. From here we jump to 5th element with value 9, and from here we will jump to the last. Input: arr = [1, 4, 3, 2, 6, 7] Output: 2 Explanation: First we jump from the 1st to 2nd element and then jump to the last element. Input: arr = [0, 10, 20] Output: -1 Explanation: We cannot go anywhere from the 1st element. Constraints: $2 \leq \text{arr.size}() \leq 10^5$ $0 \leq \text{arr}[i] \leq 10^5$</p> <p>Link: https://www.geeksforgeeks.org/problems/minimum-number-of-jumps-1587115620/1</p>	
	1	<p>Given an array of integers nums containing n + 1 integers where each integer is in the range [1, n] inclusive. There is only one repeated number in nums, return this repeated number. You must solve the problem without modifying the array nums and using only constant extra space.</p> <p>Example 1: Input: nums = [1,3,4,2,2] Output: 2 Example 2: Input: nums = [3,1,3,4,2] Output: 3 Example 3: Input: nums = [3,3,3,3,3] Output: 3</p> <p>Constraints: $1 \leq n \leq 10^5$ nums.length == n + 1 $1 \leq \text{nums}[i] \leq n$ All the integers in nums appear only once except for precisely one integer which appears two or more times.</p> <p>Link: https://leetcode.com/problems/find-the-duplicate-number/description/</p>	Take Home
	1	<p>Given two sorted arrays a[] and b[] of size n and m respectively, the task is to merge them in sorted order without using any extra space. Modify a[] so that it contains the first n elements and modify b[] so that it contains the last m elements.</p> <p>Examples:</p> <p>Input: a[] = [2, 4, 7, 10], b[] = [2, 3] Output: a[] = [2, 2, 3, 4], b[] = [7, 10] Explanation: After merging the two non-decreasing arrays, we get, [2, 2, 3, 4, 7, 10] Input: a[] = [1, 5, 9, 10, 15, 20], b[] = [2, 3, 8, 13] Output: a[] = [1, 2, 3, 5, 8, 9], b[] = [10, 13, 15, 20] Explanation: After merging two sorted arrays we get [1, 2, 3, 5, 8, 9, 10, 13, 15, 20]. Input: a[] = [0, 1], b[] = [2, 3] Output: a[] = [0, 1], b[] = [2, 3]</p>	Take Home

School of Computer Science and Engineering

Experiment List for Programming Ability and Logic Building - 1

Proposed Date	Lecture	Experiment	In Class / Take Home
		<p>Explanation: After merging two sorted arrays we get [0, 1, 2, 3].</p> <p>Constraints: $1 \leq n, m \leq 10^5$ $0 \leq a[i], b[i] \leq 10^7$</p> <p>Link: https://www.geeksforgeeks.org/problems/merge-two-sorted-arrays-1587115620/1</p>	
	1	<p>Given an array of intervals where intervals[i] = [starti, endi], merge all overlapping intervals, and return an array of the non-overlapping intervals that cover all the intervals in the input.</p> <p>Example 1:</p> <p>Input: intervals = [[1,3],[2,6],[8,10],[15,18]] Output: [[1,6],[8,10],[15,18]] Explanation: Since intervals [1,3] and [2,6] overlap, merge them into [1,6].</p> <p>Example 2:</p> <p>Input: intervals = [[1,4],[4,5]] Output: [[1,5]] Explanation: Intervals [1,4] and [4,5] are considered overlapping.</p> <p>Example 3:</p> <p>Input: intervals = [[4,7],[1,4]] Output: [[1,7]] Explanation: Intervals [1,4] and [4,7] are considered overlapping.</p> <p>Constraints: $1 \leq \text{intervals.length} \leq 10^4$ $\text{intervals}[i].\text{length} == 2$ $0 \leq \text{starti} \leq \text{endi} \leq 10^4$</p> <p>Link: https://leetcode.com/problems/merge-intervals/description/</p>	Take Home
Week: (26/01/26 to 31/01/26)	2	<p>Given three sorted arrays in non-decreasing order, print all common elements in non-decreasing order across these arrays. If there are no such elements return an empty array. In this case, the output will be -1.</p> <p><i>Note:</i> can you handle the duplicates without using any additional Data Structure?</p> <p>Examples :</p> <p>Input: arr1 = [1, 5, 10, 20, 40, 80] , arr2 = [6, 7, 20, 80, 100] , arr3 = [3, 4, 15, 20, 30, 70, 80, 120] Output: [20, 80] Explanation: 20 and 80 are the only common elements in arr1, arr2 and arr3.</p> <p>Input: arr1 = [1, 2, 3, 4, 5] , arr2 = [6, 7] , arr3 = [8,9,10] Output: [-1] Explanation: There are no common elements in arr1, arr2 and arr3.</p> <p>Input: arr1 = [1, 1, 1, 2, 2, 2], arr2 = [1, 1, 2, 2, 2], arr3 = [1, 1, 1, 1, 2, 2, 2, 2] Output: [1, 2] Explanation: We do not need to consider duplicates</p>	In Class

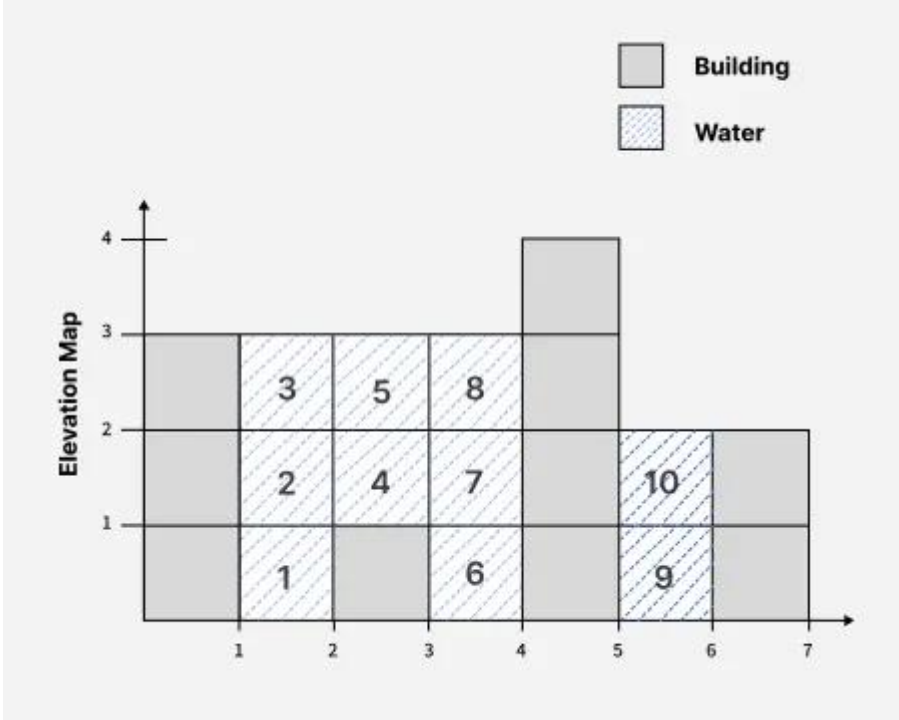
School of Computer Science and Engineering

Experiment List for Programming Ability and Logic Building - 1

Proposed Date	Lecture	Experiment	In Class / Take Home
		Link: https://www.geeksforgeeks.org/problems/common-elements1132/1	
	2	<p>Given an integer n, find its factorial. Return a list of integers denoting the digits that make up the factorial of n.</p> <p>Examples:</p> <p>Input: n = 5 Output: [1, 2, 0] Explanation: 5! = 1*2*3*4*5 = 120</p> <p>Input: n = 10 Output: [3, 6, 2, 8, 8, 0, 0] Explanation: 10! = 1*2*3*4*5*6*7*8*9*10 = 3628800</p> <p>Input: n = 1 Output: [1] Explanation: 1! = 1</p> <p>Link: https://www.geeksforgeeks.org/problems/factorials-of-large-numbers2508/1</p>	In Class
	2	<p>Given two arrays a[] and b[], your task is to determine whether b[] is a subset of a[].</p> <p>Examples:</p> <p>Input: a[] = [11, 7, 1, 13, 21, 3, 7, 3], b[] = [11, 3, 7, 1, 7] Output: true Explanation: b[] is a subset of a[]</p> <p>Input: a[] = [1, 2, 3, 4, 4, 5, 6], b[] = [1, 2, 4] Output: true Explanation: b[] is a subset of a[]</p> <p>Input: a[] = [10, 5, 2, 23, 19], b[] = [19, 5, 3] Output: false Explanation: b[] is not a subset of a[]</p> <p>Link: https://www.geeksforgeeks.org/problems/array-subset-of-another-array2317/1</p>	Take Home
	2	<p>Given an array arr[] and an integer target, determine if there exists a triplet in the array whose sum equals the given target. Return true if such a triplet exists, otherwise, return false.</p> <p>Examples:</p> <p>Input: arr[] = [1, 4, 45, 6, 10, 8], target = 13 Output: true Explanation: The triplet {1, 4, 8} sums up to 13.</p> <p>Input: arr[] = [1, 2, 4, 3, 6, 7], target = 10 Output: true Explanation: The triplets {1, 3, 6} and {1, 2, 7} both sum to 10.</p> <p>Input: arr[] = [40, 20, 10, 3, 6, 7], target = 24 Output: false Explanation: No triplet in the array sums to 24.</p>	Take Home

School of Computer Science and Engineering

Experiment List for Programming Ability and Logic Building - 1

Proposed Date	Lecture	Experiment	In Class / Take Home
		<p>Link: https://www.geeksforgeeks.org/problems/triplet-sum-in-array-1587115621/1</p>	
	2	<p>Given an array arr[] with non-negative integers representing the height of blocks. If the width of each block is 1, compute how much water can be trapped between the blocks during the rainy season.</p> <p>Examples:</p> <p>Input: arr[] = [3, 0, 1, 0, 4, 0 2]</p> <p>Output: 10</p> <p>Explanation: Total water trapped = 0 + 3 + 2 + 3 + 0 + 2 + 0 = 10 units.</p>  <p>Input: arr[] = [3, 0, 2, 0, 4]</p> <p>Output: 7</p> <p>Explanation: Total water trapped = 0 + 3 + 1 + 3 + 0 = 7 units.</p> <p>Input: arr[] = [1, 2, 3, 4]</p> <p>Output: 0</p> <p>Explanation: We cannot trap water as there is no height bound on both sides.</p> <p>Input: arr[] = [2, 1, 5, 3, 1, 0, 4]</p> <p>Output: 9</p> <p>Explanation: Total water trapped = 0 + 1 + 0 + 1 + 3 + 4 + 0 = 9 units.</p> <p>Link: https://www.geeksforgeeks.org/problems/trapping-rain-water-1587115621/1</p>	Take Home