

Lab Assignment 5 (Week 3 – Lab A, Lab B, and C)

Q1. You have been given an array of n numbers. Write a program to check whether there are any duplicate elements in the given array or not with following constraints: Constraint 1 - extra space is not a constraint, *i.e.* unlimited extra space, and Constraint 2 - you can use extra space of $O(1)$.

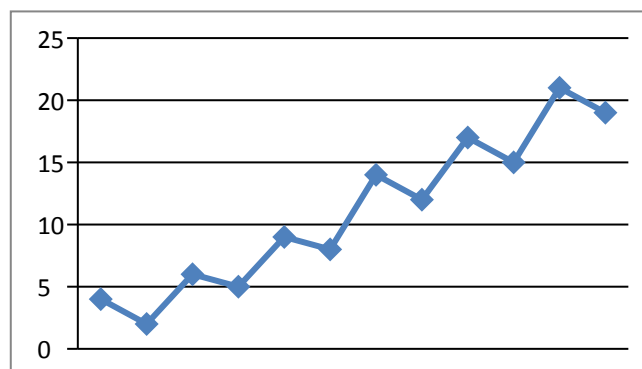
Q2. Given an array of n numbers, write a program to find the element which appears the maximum number of times in the given array? Your program must be constrained with following constraints: Constraint 1: extra space is not a constraint, *i.e.* unlimited extra space, and Constraint 2: you can use extra space of $O(1)$.

Q3. Given an array of n numbers, write a program to find the first element in the array which is repeated. For example, in the array $A = \{3, 2, 1, 2, 2, 3\}$, the first repeated number is 3 (not 2).

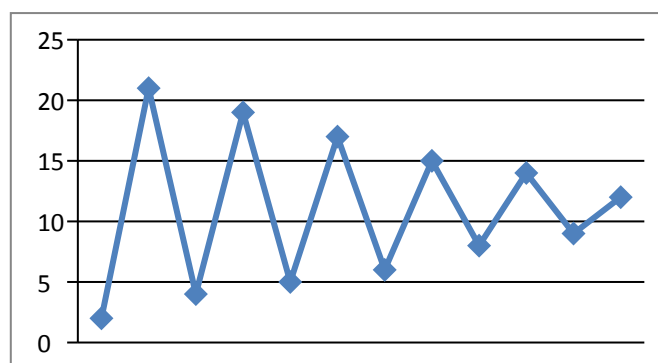
Q4. You have been given an array, ARR of randomly arranged n numbers. It is desired to arrange the elements in the array in some specific order(s) so that when plotted it presents some specific pattern. Two of such desired patterns are presented in following example:

ARR = { 2, 17, 15, 6, 21, 4, 12, 14, 5, 9, 19, 8 }

Pattern 1:



Pattern 2:



Write a program to arrange the elements of the array ARR twice so that in first arrangement it plots the Pattern 1 and in second arrangement it plots the Pattern 2.

Q5. You have been given a sorted array ARR of size M (M is very large). Elements stored in this array are 0 and 1, *e.g.* $ARR = \{0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0\}$. It is desired to compute the count of 0s in the array ARR. Propose an efficient algorithm to accomplish the task and implement the proposed algorithm.

Q6. You have been given the address of the first node of a singly linked list with N nodes. Here, each node of the linked list has two fields: an integer element and a pointer of self-type. It is desired to sort this linked list. Can you perform following, if yes then write programs:

- (a) Sort the linked list using Bubble sort with constant extra space
- (b) Sort the linked list using Quick sort with constant extra space
- (c) Sort the linked list using Merge sort with constant extra space
- (d) Sort the linked list using Selection sort with constant extra space

Q7. You have been given the address of the first node of a singly linked list. Here, each node of the linked list has two fields: an integer element and a pointer of self type. A specific pattern in elements of the successive nodes of the linked list is given as follows: element in i^{th} node (where, i is 0, 2, 4, 6, 8 ...) is bigger than the elements in $(i+1)^{\text{th}}$ node, whereas it is smaller than $(i+2)^{\text{th}}$ node. Given that extra space is not a constraint (i.e. you may use as much space as you want), which one of the following sorting technique you will apply to sort the given linked list: bubble sort, merge sort, selection sort.

Q8. Hash table and hash functions in hashing are used to store the items/elements in such a way that it will be easier to find the elements later using the hash function. Hashing can be open or closed. In open hashing keys/elements are stored in linked list attached to cells/slots of the hash table whereas, all keys/elements in closed hashing are stored in the hash table itself. Collision is one of the major problems in hashing where two or more than two elements share the same hash value and hence needed to be resolved. Linear probing, quadratic probing, re-hashing, chaining, etc. are used to implement the hashing and resolve the collision. Write programs to implement hashing using (a) linear probing, (b) quadratic probing, (c) re-hashing, and (d) chaining and store following elements in each scenario: 5, 2, 15, 25, 11, 19, 65, 23, 34, 44, 64, 74, 73, 2, 15, 29, 55, 65, 26, 64, 38, 12, 16, 49, 11, 78, 65, 63, 57, 42 Further, perform following operations:

- (a) Search an element 44
- (b) Search an element 94
- (c) Search an element 65 along with its count
- (d) Remove all duplicate entries
- (e) Search an element which appears maximum number of times
- (f) Search two elements, X and Y whose sum is equal to user inputted number Z

Q9. Given an array of n elements, find two elements in the array, such that, their sum is equal to the given element K .

Q10. Given an array of n elements, find three elements in the array, such that, their sum is equal to the given element K .