

DATA STRUCTURES & ALGORITHMS
ASSIGNMENT#1
DEADLINE: for ClassId: 100818 BY 28th Feb, 2019
for ClassId: 100819 BY 01st Mar, 2019

Instructions:

- Use C# language for writing the code.
- Do not use any built-in method or class such as array.Sort, array.resize, array.Length etc etc.
- Do not hard code any value unless explicitly mentioned.
- Briefly explain the Big-O complexity for each solution.
- Provide printed copy for codes and handwritten version for the algorithms.
- Do not provide any extra information.

| |
|-------------------------------|
| ONE DIMENSIONAL ARRAYS |
|-------------------------------|

Q1- Provide the C# implementation of the following algorithms/pseudocodes.

- a) Traversing a linear array (Slide#17)
- b) Inserting into a linear array – all cases (as discussed in lecture)
- c) Deletion from a linear array – all cases (as discussed in lecture)
- d) Linear searching (Slide#48)
- e) Binary searching (as discussed in lecture)
- f) Bubble Sort (as discussed in lecture)
- g) Selection Sort (as discussed in lecture)
- h) Insertion Sort (as discussed in lecture)

Q2- What should the following array look like after applying Bubble Sort (ascending order)? Show each step clearly. Mention the total number of swapping required.

4, 3, 9, 3, 1

Q3- What should the following array look like after applying Selection Sort (descending order)? Show each step clearly. Mention the total number of swapping required.

4, 3, 9, 3, 1

Write an algorithm (modified selection sort) for the working you have used to sort the above array in descending order.

Q4- What should the following array look like after applying insertion sort (ascending order)? Show each step clearly. Mention the total number of insertions required.

4, 3, 9, 3, 1

Q5- Consider the following array already sorted in descending order.

| | | | | | | | | | |
|----|----|----|----|----|----|----|---|---|---|
| 68 | 37 | 25 | 22 | 17 | 15 | 11 | 9 | 5 | 3 |
|----|----|----|----|----|----|----|---|---|---|

- a) Using binary search algorithm,
 - search any one of the possible values that forces the algorithm to take the maximum time (worst case). Show all the working properly.
 - search the value that forces the algorithm to take the minimum time (best case). Show all the working properly.
- b) Calculate the Big-O complexity for the given input using the derived formula.
- c) Write an algorithm (Modified binary search) for the working you have shown to solve part (a).

Q6- A left rotation operation on an array of size n shifts each of the array's elements 1 unit to the left. For example, if 2 left rotations are performed on array [1,2,3,4,5] , then the array would become [3,4,5,1,2]

When we perform 4 left rotations, the array undergoes the following sequence of changes:

$[1, 2, 3, 4, 5] \rightarrow [2, 3, 4, 5, 1] \rightarrow [3, 4, 5, 1, 2] \rightarrow [4, 5, 1, 2, 3] \rightarrow [5, 1, 2, 3, 4]$

Write an algorithm for a procedure that given an array of n integers and a number, d, performs d left rotations on the array. Then prints the updated array as a single line of space-separated integers.

Q7- Write an algorithm for a procedure that given a sequence(array) of numbers, sorts, even numbers in ascending order, odd numbers in descending order and places all even numbers in the initial part of an array then odd numbers. Example: 2, 5, 1, 0, 4, 7, 9, 3, -2, 10, 20, 15 The answer is: -2, 0, 2, 4, 10, 20, 15, 9, 7, 5, 3, 1.

Q8- There is a collection of input integers and a collection of query integers. For each query integer, determine how many times it occurs in the list of input integers.

For example, given input integers = [1,1,2] and queries = [1,2,3] we find 2 matching of 1, 1 of 2 and 0 of 3. For each query, we add an element to our return array, results = [2,1,0]

- Select an appropriate searching algorithm to solve this problem. Also mention the reason.
- Use the selected algorithm to write a procedure (algorithm) named MatchingIntegers(). The procedure must return an array of integers representing the frequency of occurrence of each query integer in integers.

MatchingIntegers() has the following parameters:

- integers - an array of integers to search
- queries - an array of query integers

Q9- There is a collection of input strings and a collection of query strings. For each query string, determine how many times it occurs in the list of input strings.

For example, given input strings = [ab,ab,bc] and queries = [ab,bc,abc] we find 2 matching of ab, 1 of bc and 0 of abc. For each query, we add an element to our return array, results = [2,1,0]

- Which of the two sorting algorithms best solves this problem. Why have you selected this algorithm?
- Use the selected algorithm to write a procedure (algorithm) named MatchingStrings(). The procedure must return an array of integers representing the frequency of occurrence of each query string in strings.

MatchingStrings() has the following parameters:

- strings - an array of strings to search
- queries - an array of query strings

TWO DIMENSIONAL ARRAYS

Q10- Write a program containing three functions named 'Input_Matrix', 'Transpose_Matrix' and 'Display_Transpose' to enter, calculate and display the transpose of a two dimensional matrix. The dimensions of the matrix must be taken as input. Use appropriate parameters and return type.

Q11- Suppose you have the following matrices:

$$M1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix} \text{ and } M2 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

Write an algorithm for a procedure that given two matrices calculates and returns their product.

$$M1 \times M2 = \begin{pmatrix} 70 & 80 & 90 \\ 158 & 184 & 210 \end{pmatrix}$$

Q12- Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix is shown below:

```
1 2 3
4 5 6
9 8 9
```

The left-to-right diagonal = 1+5+9=15.

The right to left diagonal = 3+5+9 = 17

Their absolute difference is |15-17| = 2

Provide an algorithm for a procedure DiagonalDifference(). It must return an integer representing the absolute diagonal difference.

DiagonalDifference takes the following parameter:

- arr: a 2D array of integers.

Q13- A local zoo wants to keep track of how many pounds of food each of its three monkeys eats each day during a typical week. Stores this information in a two dimensional 3 × 7 array, where each row represents a different monkey and each column represents a different day of the week. First let the user input the data for each monkey then it should create a report that includes the following information:

- Average amount of food eaten by each monkey for the whole week.
- The least amount of food eaten during the week by any one monkey.

- The greatest amount of food eaten during the week by any one monkey.

Write a C# program to solve the problem.

THREE DIMENSIONAL ARRAYS

Q14- Write an algorithm that calculates the total score for students in a class. Suppose the scores are stored in a three-dimensional array named `scores`. The first index in `scores` refers to a student, the second refers to an exam, and the third refers to the part of the exam. Suppose there are 7 students, 5 exams, and each exam has two parts--the multiple-choice part and the programming part. So, `scores[i][j][0]` represents the score on the multiple-choice part for the *i*'th student on the *j*'th exam. Your algorithm should display the total score (all exams collectively) for each student.