CMP-210 Data Structures and Algorithms
BS Fall 2018
Assignment 1

Roll-No: _ _ _ _ _ _ _ _ _

Name: _ _ _ _ _ _ _ _ _ _

Issue Date: 3-Mar-2020
Submission Date: 9-Mar-2020
Marks: 54

Objective:

- It will test your ability to calculate the Time/Space equation.
- And it will force you to get your hands dirty: It's good for you ☺.

Challenge-A:

Part-A

Give time equation of the following code segment. (see the example posted on piazza about how to fill the table for step count)

Snippet-A

void fun(int n)	**Step Count for each statement**
{	
for (int i = 1; i <= n/2; i++)	
{	
if (i%1 == 0)	
{	
for (int j = 1; j < n/2; j++)	
{}	
}	
else if (i%2 != 0)	
{	
for (int j = 1; j < n*n; j++)	
{}	
}	
}	
}	
Final simplified equation:	

Snippet-B

void fun(int n)	**Step Count for each statement**
{	
int x = 0;	
for (int i = n/2 ; i < n; i++)	
{	
if (i % 5 == 0)	
break;	
for(int j = 1; j < n; j += 2)	
{	
x++;	
x *= 2;	
}	
}	
}	
Final simplified equation:	

Snippet-C: Write worst case time bound of the following snippet

`x = 2;`
`k = 1;`
`while ( k <= n )`
`{`
`    x = x + x;`
`    k = x * x;`
`}`
	BigO

Challenge-B:

An array is called perfect if the elements present in it are in the range of 0 to N-1, where N is the size of array and In this array no number is missing; means all the values in the range 0 to N-1 will be present in the array.
But we have a problem, every time we receive an array there is one element missing in it or in other words it contains zero as value in place of the missing element.
Your task is to find the missing element in *O*(N) time and *O*(1) space.

Challenge-C:

Given a string consisting of English alphabets (case Sensitive), find the maximum length of string which forms a palindrome.
For example, for the string "xbazabaazzy", the largest length can be 9 and one of such palindrome is "azabzbaza". Do it *O*(N) time and *O*(1) space bound.

Challenge-D:

Given an array of integers of size N, the ceiling of x is the smallest element in the set greater than or equal to x, and the floor is the largest element less than or equal to x. Suppose you have an array of N items in ascending order. Give a *O*(log N) time and *O*(1) space algorithm to find the floor of x in the array.

Challenge-E:

Modified 2 color sort problem i.e. you are given an array of integers containing only 0s and 1s.You have to place all the 0s in even position and 1s in odd position. And if suppose, count of 0's exceed the count of 1's or vice versa then keep them untouched. Do it *O*(N) time but do it in ONE PASS (reading the array only once) and O(1) space.
For Example :

Input Array: {0,1,1,0,1,0,1,0,1,1,1,0,0,1,0,1,1}
Output Array: {0,1,0,1,0,1,0,1,0,1,0,1,0,1,1,1,1}

Challenge-F:

Given a string, write the code to remove the duplicate elements present in them. For example, if the given string is "HelloWorld", then, the output has to be "HeloWrd". As another example, if given string is "AbRAcaDaBRA", then, the output has to be "AbRcaDB".
Do it in O(N) time and *O*(1) space.