UNIVERSITY OF THE PUNJAB

B.S. in Computer Science: Fourth Semester – Fall 2023

Paper: Data Structure & Algorithm **Course Code:**CC-212 **Time:** 3 Hrs. **Marks:** 60

THE ANSWERS MUST BE ATTEMPTED ON THE ANSWER SHEET PROVIDED

Q.1. Answer the following short questions: (5x6=30)

1. 3A three-dimensional array (having lower bound zero) $A[U_1][U_2][U_3]$ stored in a row-major-order with base address 100. The dimensions of the array are:

- **$U_1 = 5$, $U_2 = 3$, and $U_3 = 10$**

You are required to calculate the address of location $A[i][j][k]$, where i = 3, j = 1, and k = 6.

Also, suppose that each cell occupies 2 bytes in memory.

Formula to Find the Offset of A [i][j][k]

Offset= $i(U_2 * U_3)+j(U_3)+k$

= 3(3*10)+1(10)+6

=3(30)+10+6

=90+10+6=106

Formula to Find the Base Address

Address= BaseAddress +(Offset*size_of_each_element)

= 100+ (106*2)

= 100+ 212

=312

2. Obtain the step count of the following code segment:

```
for (i = 0; i < n; i++)
temp;
```

Operation	Steps

`Initialization`	`1`
`Condition checks`	`n + 1`
`Body (temp;)`	`n`
`Increment (i++)`	`n`
`Total`	**`3n + 2`**

3. The running time of a program is $4N^3 + 5N^2 + 6N + 7$, where N is the input size. What are the upper and lower bounds for the growth rate of the program? Do not forget to mention the constants (c and n_0).

Upper Bound (Big-O)

$T(N) = 4N^3 + 5N^2 + 6N + 7$

$\leq 4N^3 + 5N^3 + 6N^3 + 7N^3$ [since $N^2 \leq N^3$, $N \leq N^3$, $1 \leq N^3$ for $N \geq 1$]

$= (4 + 5 + 6 + 7) * N^3$

$= 22N^3$

c=22

And $n_0 = 1$

$T(N) = O(N^3)$ as highest polynomial is N^3

Lower Bound (Big-Omega)

By Ignoring lower degree Polynomial we have **$T(N) = 4N^3$**

c = 4

$n_0 = 1$

$T(N) = \Omega(N^3)$

4. An algorithm takes 10 seconds for an input size of 1000.

How long will it take for input size of 100 if the running time is

5N?

Let's assume: **The time taken T(N) is given by T(N)=c*5(N)**

Given:
T(1000)=c*5*1000=10 seconds
5000c=10
=0.002

Now find T(100):

$$c = \frac{10}{5000}$$

T(100)=0.002*5*100=0.002*500=**1 second**

5. Convert A * ((B - C) - D) into its equivalent pre fix form.

A * ((B - C) - D)
A * (-B C - D)
A * --B C D
*A --B C D

Q.2. Answer the following questions: (3x10=30)

1. Sort the following array into increasing order using Bubble Sort.
Clearly show the contents of the array after each iteration of the outer loop.
You will get NO credit if it is not clear that you used Bubble Sort to sort the given array.

39, 37, 38, 36, 30 ,32, 24, 35,31,33

First Iteration:

39 is Compared with all of its adjacent vertices one by and then moved if 39 is greater than the adjacent vertice on the right side. Comparison is represented in green color.

39	37	38	36	30	32	24	35	31	33

37	39	38	36	30	32	24	35	31	33
37	39	38	36	30	32	24	35	31	33
37	38	39	36	30	32	24	35	31	33
37	38	39	36	30	32	24	35	31	33
37	38	36	39	30	32	24	35	31	33
37	38	36	39	30	32	24	35	31	33
37	38	36	30	39	32	24	35	31	33
37	38	36	30	39	32	24	35	31	33

37	38	36	30	32	39	24	35	31	33
37	38	36	30	32	39	24	35	31	33
37	38	36	30	32	24	39	35	31	33
37	38	36	30	32	24	39	35	31	33
37	38	36	30	32	24	35	39	31	33
37	38	36	30	32	24	35	39	31	33

37	38	36	30	32	24	35	31	39	33

37	38	36	30	32	24	35	31	39	33

37	38	36	30	32	24	35	31	33	39

After First Iteration we have:

37	38	36	30	32	24	35	31	33	39

Iteration 2:

37 is compared with 36. Since 37 > 36, they are swapped. Then 37 is compared with the next element until it Encounters a number greater than itself

37	36	30	32	24	35	31	33	38	39

Iteration 3:

In this Iteration 37 is compared with 36. As 37 is greater then it will be compared with the rest of elements on the right side Until it encounters a number greater than itself.

36	30	32	24	35	31	33	37	38	39

Iteration 4:

30	32	24	35	31	33	36	37	38	39

Iteration 5:

As 32 is greater than 30 so 32 is a bubble now

30	24	32	35	31	33	36	37	38	39

Now when **35 is greater than 32**, 35 will be the bubble now. And will be compared with all others. Until it encounters a number greater than itself

30	24	32	35	31	33	36	37	38	39

30	24	32	31	33	35	36	37	38	39

Iteration 6:

30 is bubble Now

30	24	32	31	33	35	36	37	38	39

24	30	32	31	33	35	36	37	38	39

32 is bubble now

24	30	31	32	33	35	36	37	38	39

24	30	31	32	33	35	36	37	38	39

Iteration 7:

30 is bubble

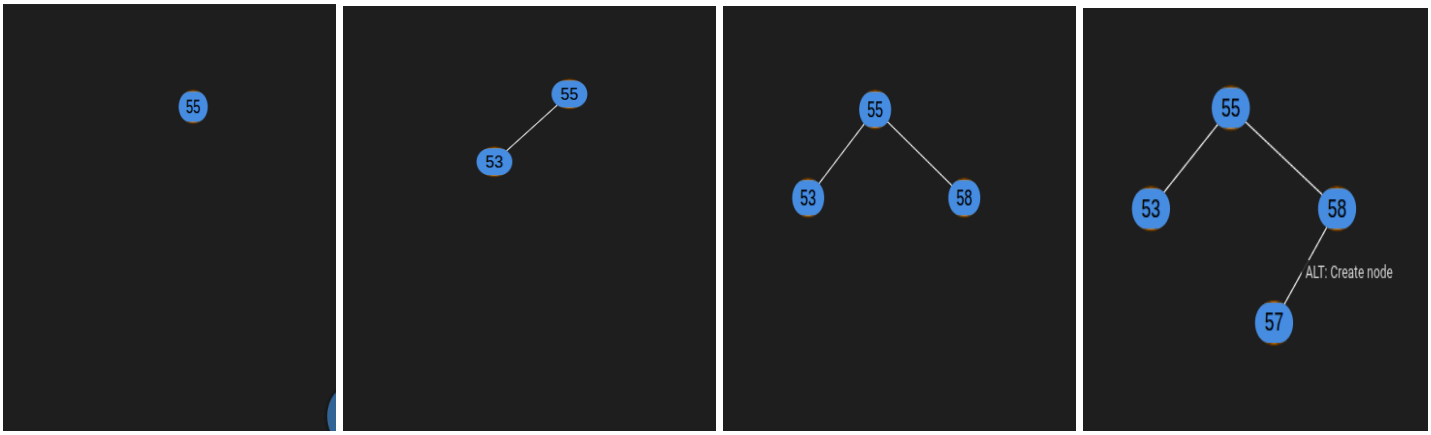
24	30	31	32	33	35	36	37	38	39

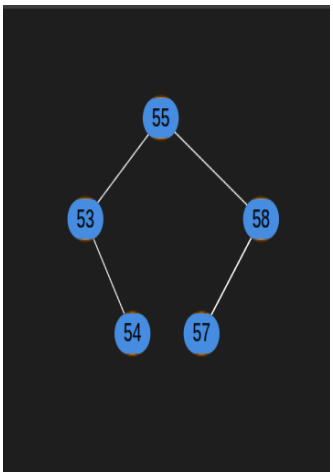
24	30	31	32	33	35	36	37	38	39

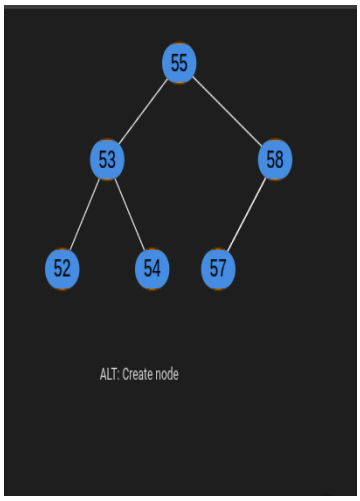
2. Draw a Binary Search Tree (BST) when the following values are inserted one-by-one (in the given order from left to right) into an initially empty BST:

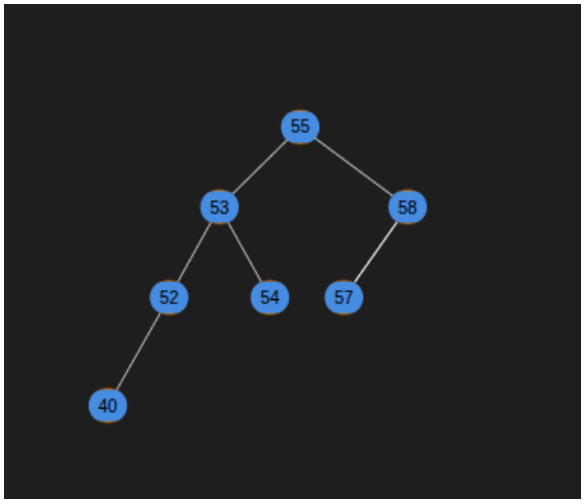
Values:

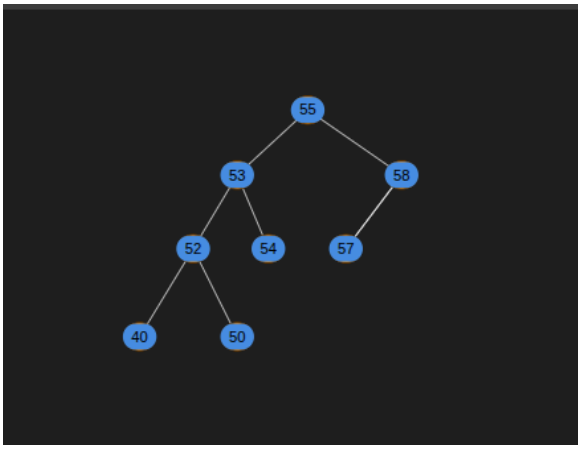
55, 53, 58, 57, 54, 52, 40, 50, 59, 51

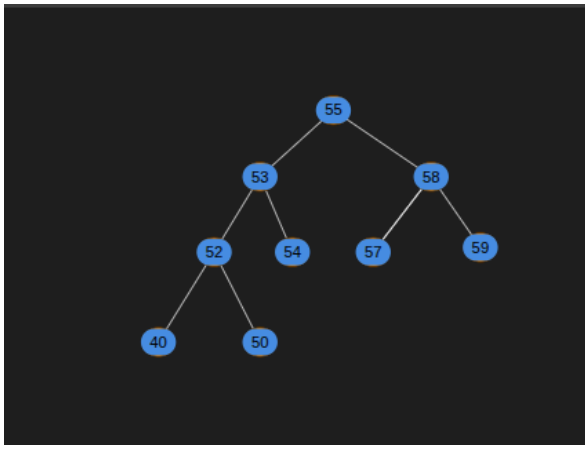




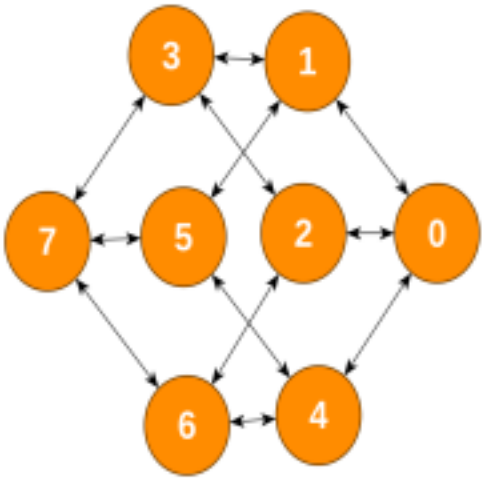








3. Given the following directed graph, G,



write the sequence of vertices after performing Breadth-First Search (BFS) starting from the source vertex A. Also, show the resulting BFS tree.If there is more than one choice of vertices at any given point, traverse them in alphabetical order.

Read More…