DSA 2023 May

(Consider that answered only for the questions that belongs to DSA II in L2S2 in new syllabus)

Questions 01

(a)

Recursion	**Iteration**
Small code size.	Larger code size
Space complexity is higher than iterations. Because each function call adds a new layer to the stack.	Relatively lower space complexity.
Higher time complexity	Lower time complexity

(e)

1. Select a pivot
2. Partitioning
   - Rearrange the elements of the array as,
     i) The elements that less than pivot is come before it.
     ii) The elements that higher that pivot is come after it.
3. Recursively sort sub arrays

(f)

1. First element
2. Middle element
3. Last element

(g)

46	29	37	55	23	17	20	5

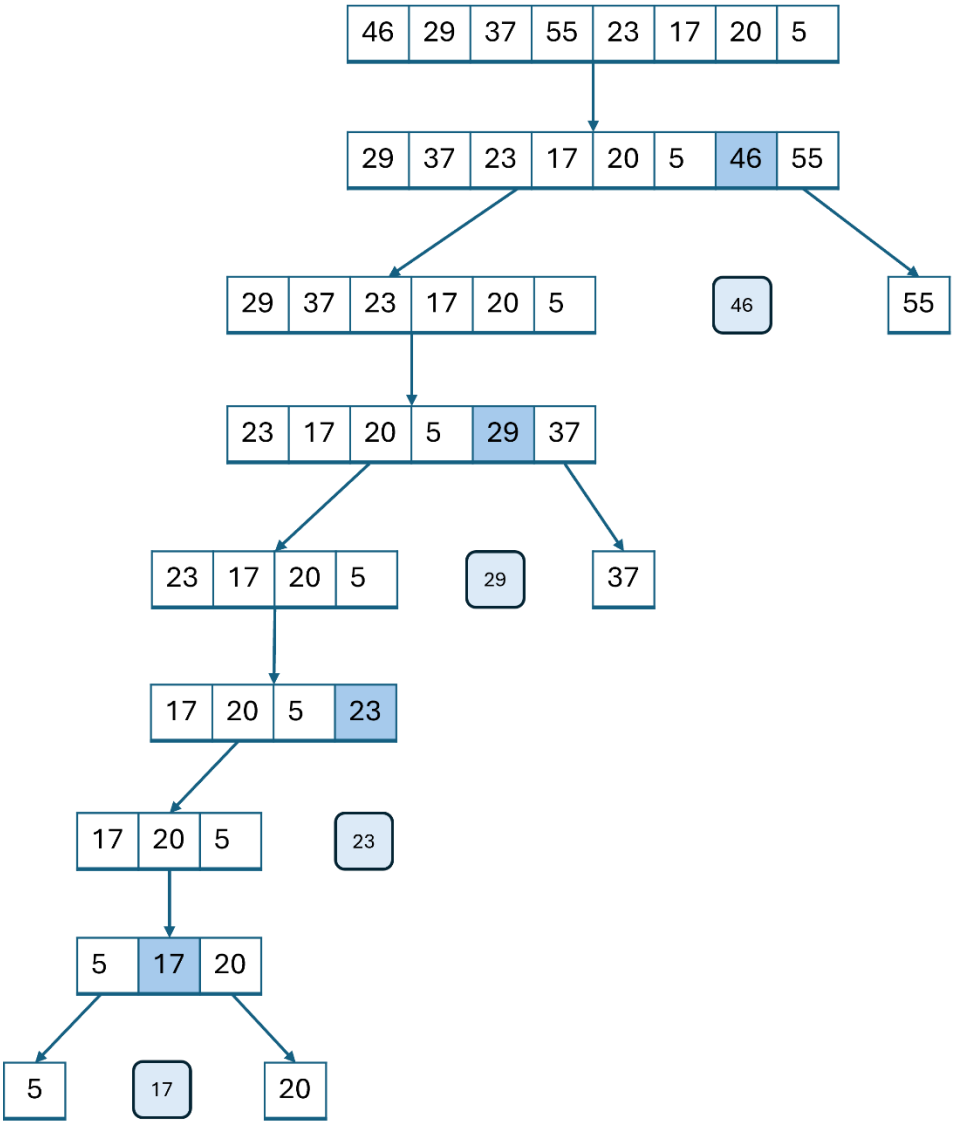
29	37	23	17	20	5	46	55

23	17	20	5	29	37	46	55

17	20	5	23	29	37	46	55

5	17	20	23	29	37	46	55

Quicksort execution can be described as follows:



Question 03

(a)

Static Memory Allocation

- The process of allocating memory for variables during compile time. The memory allocation is predictable and has a faster execution time. But once allocated memory for variables as static memory allocation it can't be changed.

Dynamic Memory Allocation

- Allocate memory during runtime. This memory allocation is slower but it flexible (Can be changed after allocating memory)

(c)

i) Time Complexity

Time complexity measures the amount of time that an algorithm takes to execute its code according to length of the input.

ii) Space Complexity

The space complexity measures the amount of space that an algorithm required to execute the code.

(d)

```
void testFunction(int n){
    for (int i = 0; i < n; i++){ ---------------------------------------------- O(n)
        for (int j = 0; j < 10; j++){ ---------------------------------------- O(10) -> O(n)
            for (int k = 0; k < n; k++){  ---------------------------- O(n)
                for (int m = 0; m < 20; m++){ ---------- O(20) -> O(n)
                    System.Out.println("*"); --------- O(1)
                }
            }
        }
    }
}
```

All the for loops are in each other,

Therefore,

Time Complexity = $O(n * n * n * n + 1)$

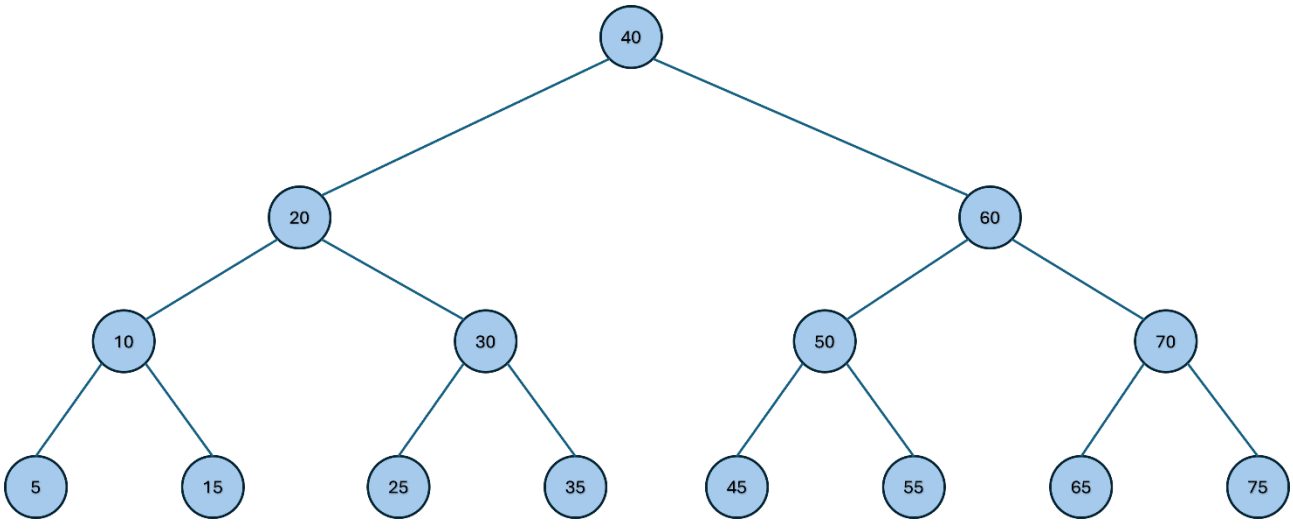
$= O(n^4 + 1)$

$= O(n^4)$

Question 04

(a)

i)

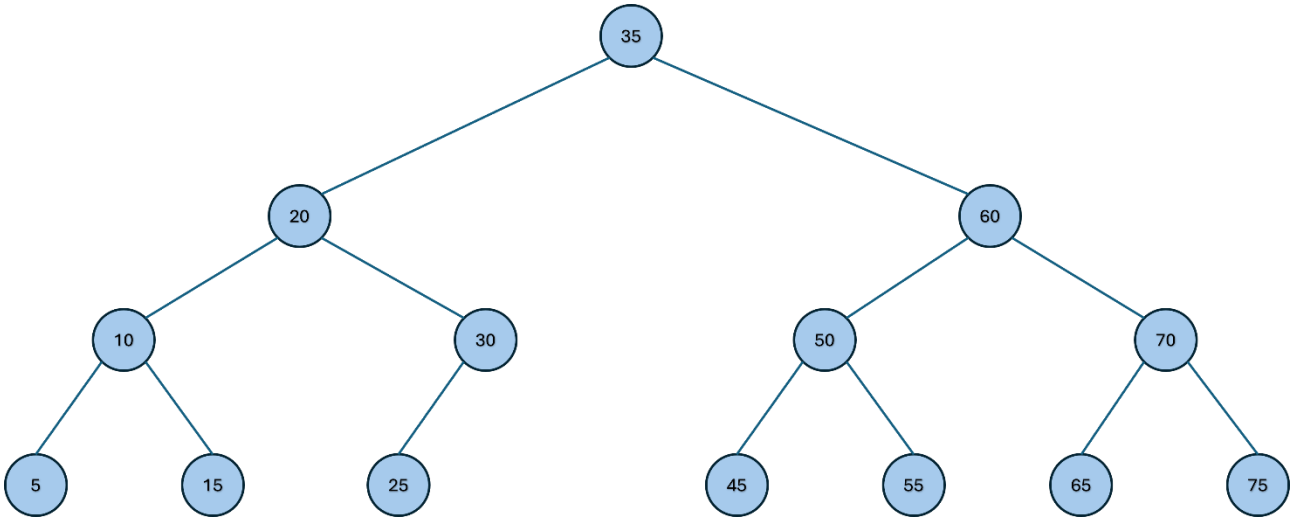


ii)

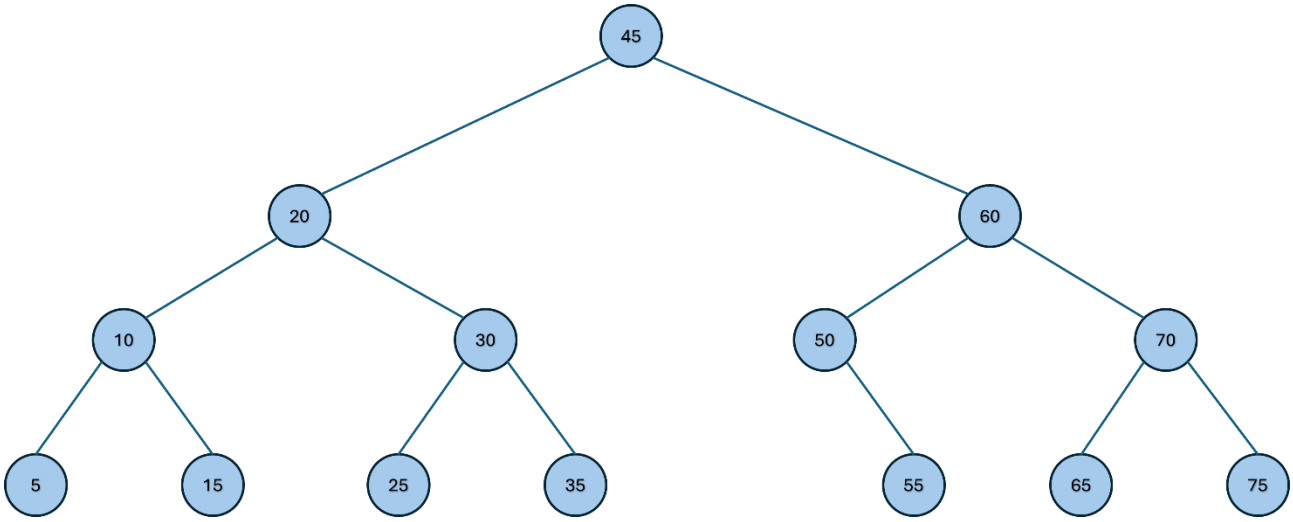
Traversal Name	Order														
Pre-order	40	20	10	5	15	30	25	35	60	50	45	55	70	65	75
In-order	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
Post-order	5	15	10	25	35	30	20	45	55	50	65	75	70	60	40

iii)

Using predecessor:



Using successor:



(b)

A directed acyclic graph is a graph which has directed edges and does not having at least one cycle within the graph.

(c)

Adjacency Matrix

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 2 & 4 & 5 & 0 \\ 0 & 0 & 0 & 7 & 3 \\ 2 & 0 & 0 & 6 & 0 \\ 0 & 3 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

(d)

Breadth First Search Order:

0	1	3	2	5	6	4