

Question 1: Characters Median and Mode

Run the java file **MedianMode.java** to see the output.

For calculating Median:

Time Complexity: $O(n \log n)$

Space Complexity: $O(1)$

For calculating Mode:

Time Complexity: $O(n)$

Space Complexity: $O(n)$

Question 2: Where is the zero?

Run the java file **Zero.java** to see the output.

Approach:

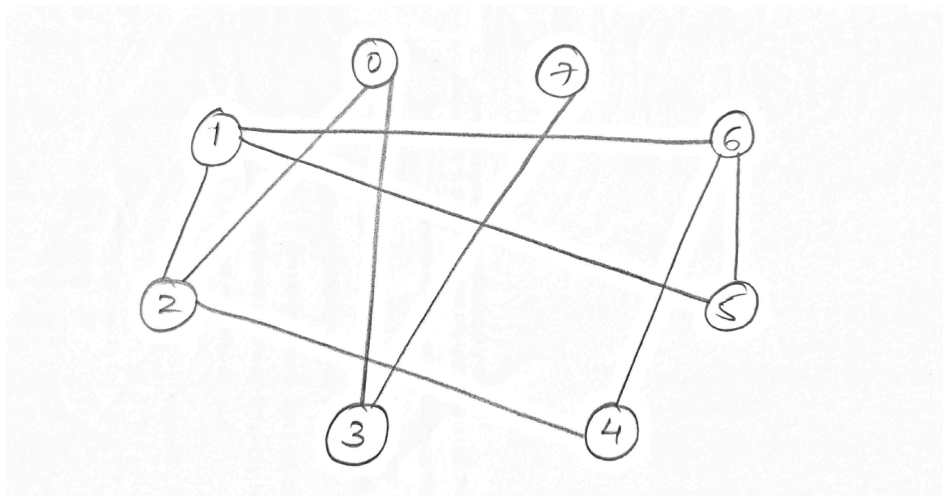
We use approach similar to binary search, where we compare the middle element of array B with that of A. If both the elements are same, we compare middle element towards the right half of array B otherwise towards the left half to find the index of zero.

Time Complexity: $O(\log n)$

Space Complexity: $O(\log n)$

Question 3: Social Graph

For 3a the graph is as shown below.



Run the java file **Socialgraph.java** to see the output for 3b, 3c and file **Dijkstra.java** for 3d.

BFS Traversal: 0 2 3 1 4 7 5 6

DFS Traversal: 0 2 1 5 6 4 3 7

Shortest distance from vertex 0 to 5 is 3

Question 4: Topological Sorting

Run the java file **Graph.java** to see the output for topological sorting.

Approach:

Here we start with the node which has no incoming edges i.e.: node with zero indegree. Such node has no dependency on any other node, so we start with 'e' in our case and then keep on traversing nodes and removing dependencies.

Topological sorting order: e g f b d c a

Question 5: Matching Classes

1. $O(n)$
2. $O(n \log n)$
3. $O(n^2)$
4. $O(\log n)$
5. $O(n)$

Question 6: Multi graph to Simple graph

Run the java file **MultiGraph.java** to see the output for adjacency list before and after sorting it to simple graph representation. The simple graph adjacency list has no loops and cycles between two nodes.

Question 7: Lombardi Graphs

Run the java file **Lombardi.java** to see the output for number of shortest paths for a particular graph example which is included in the code. The weights for all the edges in the graph is one. We have graph with 8 edges and we find the number of shortest paths from vertex 0 to 7.