1. Find Cycle in Directed Graph
    a. https://leetcode.com/problems/course-schedule/

```
bool isCycle = false;
  void dfs(int u)
  {
      visited[u] = true;
      revstack[u] = true;
      for(auto v : list[u])
      {
          if(!visited[v])
              dfs(v);
           else if(revstack[v])
              isCycle = true;
      }
      revstack[u] = false;
  }
```

2. Find Cycle in Undirected Graph
    a. geeksforgeeks.org/detect-cycle-undirected-graph/
    b. Idea is while traversing using dfs if adjacent node is already visited and other than parent than its cycle

3. Topological Sort
    a. DFS
        i. Add last line of dfs algo => stack.push(u)
        ii. Pop all element of stack to vector this vector contains topological order
    b. BFS
        i. https://www.youtube.com/watch?v=_LuIvEi_kZk
        ii. Pick all the vertices with in-degree as 0 and add them into a queue
        iii. Remove a vertex from the queue
            1. Decrease in-degree by 1 for all its neighboring nodes.
            2. If in-degree of a neighboring nodes is reduced to zero, then add it to the queue.
            3. Increment visited_nodes by 1
        iv. If visited_nodes != total_nodes
            Cycle in graph [since nodes which are in cycle will never have indegree zero they will never come in queue]
    c. To find longest topological order length
        i. https://leetcode.com/problems/longest-increasing-path-in-a-matrix/discuss/288520/BFS-Implemented-Topological-Sort
        ii. USE bfs version 2
            1. while(! q.empty())
                Lev++
                Size = q.size()
                for(i=0;i<size;i++)
                // //

4. Euclier Path
    a. https://leetcode.com/problems/reconstruct-itinerary/

```
 void dfs(string u,  unordered_map<string, multiset<string>> & list,
vector<string> &ans){
            while(list[u].size()){
                string next = *list[u].begin();
                list[u].erase(list[u].begin());
                dfs(next, list, ans);
            }
            ans.push_back(u);
    }
Return reverse(ans.begin(),ans.end())
```

5. Strongly Connected Components
    a. https://www.interviewbit.com/problems/strongly-connected-components/
    b. Topological sort
    c. One by one take output of a.
        i. Apply dfs of each and store in vector

6. Floyd warshal
    a. Make diagonal 0 and set INF 9999 (not INT_MAX cause overflow)

```
for (k = 0; k < V; k++)
   for (i = 0; i < V; i++)
       for (j = 0; j < V; j++)
       {
           if (dist[i][k] + dist[k][j] < dist[i][j])
               dist[i][j] = dist[i][k] + dist[k][j];
       }
   }
}
```

7. Bipartite Graph
    a. If graph is 2 colorable than it is Bipartite
        i. https://leetcode.com/problems/is-graph-bipartite/
    b. To color k color to graphss
        i. https://leetcode.com/problems/flower-planting-with-no-adjacent/

8. Largest Distance between any two node in n - array tree
    a. interviewbit.com/problems/largest-distance-between-nodes-of-a-tree/
    b. Find farthest node x from any of the node u of tree - this node will be one of the end point of largest diameter - using bfs / dfs
    c. Now find farthest node y from given node x
    d. Ans will be distance of y from x

9. Stepping Numbers
    a. https://www.interviewbit.com/problems/stepping-numbers/

                5
          54   56
       543 545   565   567

    START from 0 to 9 generate each stepping no

Boundary case if last digit is 0 or 9

10. Find the centroid of a tree - undirected graph which has a almost same dis to all nodes
    a. https://leetcode.com/problems/minimum-height-trees/
    b. Delete leaves until there are utmost 2 nodes left

11. No of islands where same row or same columns are connected
    a. https://leetcode.com/problems/most-stones-removed-with-same-row-or-column/

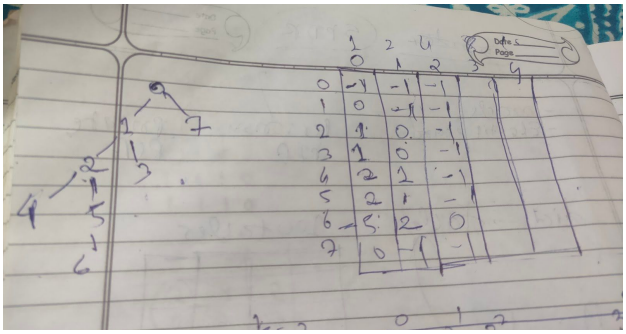
12. Number of distinct islands where same islands are not counted
    a. https://www.geeksforgeeks.org/find-the-number-of-distinct-islands-in-a-2d-matrix/
    b. Use set of vector of points
    c. Do a dfs on each components and store its shape on set
    d. To find shape of component subtract it from base point
    e. Return set size

13. Parallel BFS in Graph
    a. https://leetcode.com/problems/rotting-oranges/
    b. Start with multiple 0 as a source and put these into queue
    c. While loop

        Update adjacent point if dis[adjx][adjy] > dist[x][y] + 1 and push into queue

Binary lifting

14. Kth ancestor in binary tree
    a. https://leetcode.com/contest/weekly-contest-193/problems/kth-ancestor-of-a-tree-node/
    b. Each no can be represented in sum of power of 2
    c. 4 ancestor is 2nd ancestor of 2nd ancestor



15. LCM using logn per query and nlogn precomputations
    a. https://www.youtube.com/watch?v=02zM-QoKoPg
    b. https://www.codechef.com/LRNDSA08/submit/ENOC1

Other

1. https://www.interviewbit.com/problems/word-search-board/
2. https://www.interviewbit.com/problems/word-ladder-i/
3. https://www.interviewbit.com/problems/sum-of-fibonacci-numbers/

    a. Greedy works heres
4. https://www.interviewbit.com/problems/clone-graph/
5. https://leetcode.com/problems/find-eventual-safe-states/
6. https://leetcode.com/problems/is-graph-bipartite/
7. https://leetcode.com/problems/as-far-from-land-as-possible/
8. https://leetcode.com/problems/shortest-path-with-alternating-colors/