Experiment No. - 9

Student Name: Vivek Kumar **UID: 21BCS8129**
Branch: BE-CSE(LEET) **Section/Group: 20BCS-ST-801/B**
Semester: 6th **Date of Performance: 02/05/2023**
Subject Name: Competitive coding - II **Subject Code: 20CSP-351**

1. **Aim/Overview of the practical:**

   Q.1 All Path from Source to Target.

   https://leetcode.com/problems/all-paths-from-source-to-target/

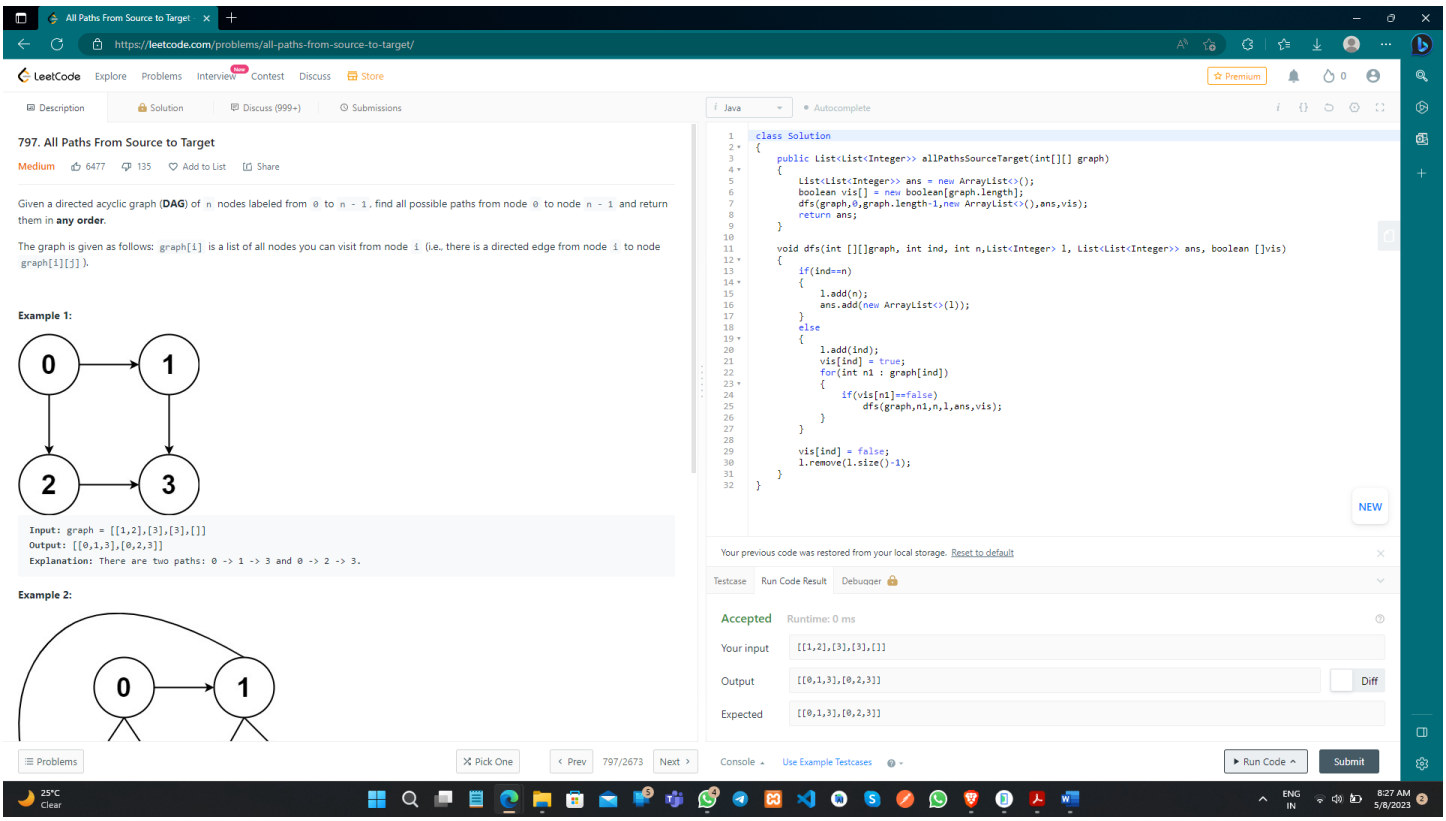
2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome
3. **Objective:**
   - To understand the concept of Backtracking.
   - To implement the concept of All path from source to target.
4. **Code:**

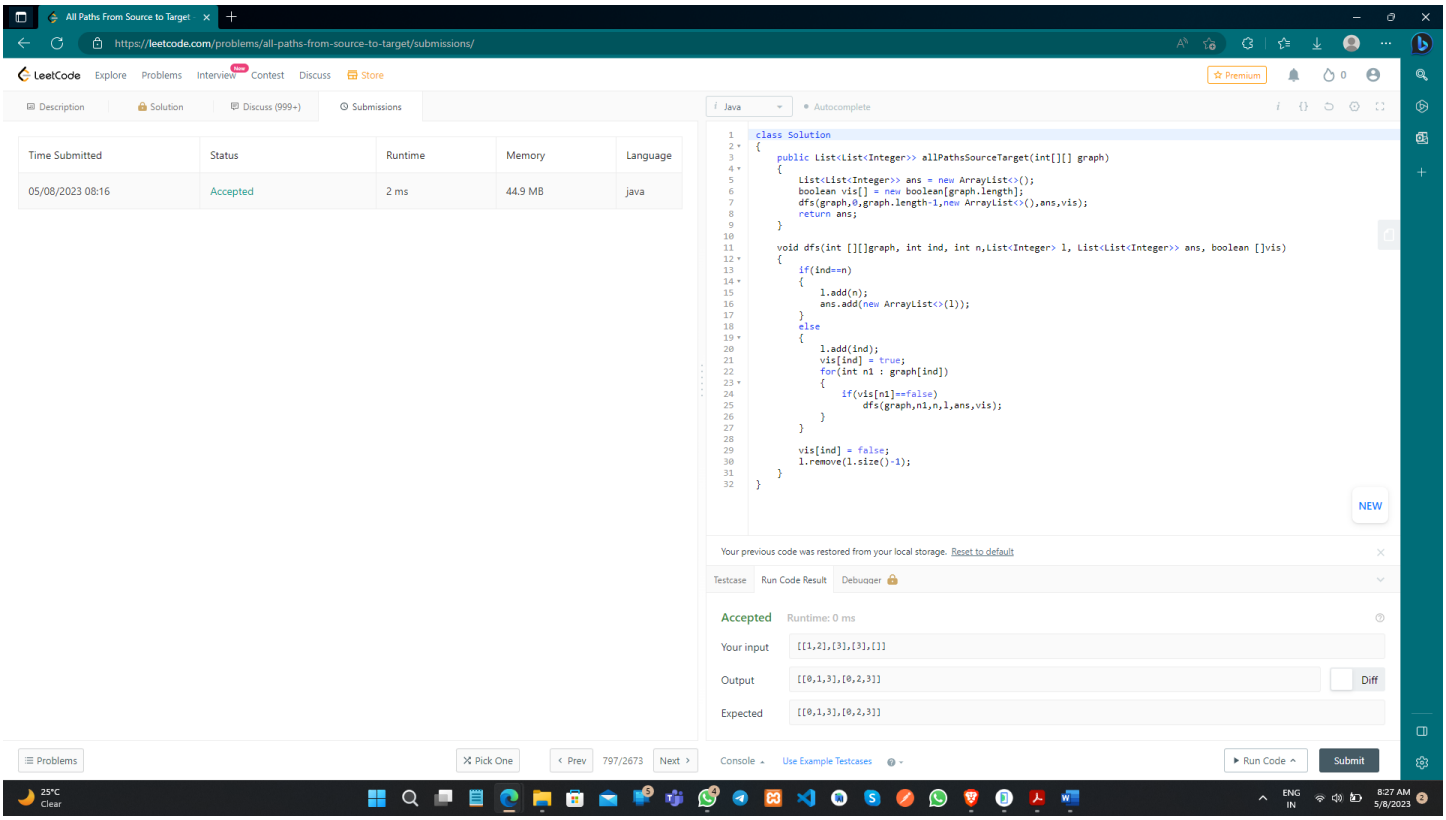
```
class Solution{
   public List<List<Integer>> allPathsSourceTarget(int[][] graph) {
      List<List<Integer>> ans = new ArrayList<>();
      boolean vis[] = new boolean[graph.length];
      dfs(graph,0,graph.length-1,new ArrayList<>(),ans,vis);
      return ans;
   }

   void dfs(int [][]graph, int ind, int n,List<Integer> l, List<List<Integer>> ans, boolean []vis){
      if(ind==n){
         l.add(n);
         ans.add(new ArrayList<>(l));
      }
      Else{
         l.add(ind);
         vis[ind] = true;
         for(int n1 : graph[ind]){
            if(vis[n1]==false)
               dfs(graph,n1,n,l,ans,vis);
         }
      }

      vis[ind] = false;
      l.remove(l.size()-1);
   }
}
```

5. Result/Output/Writing Summary:

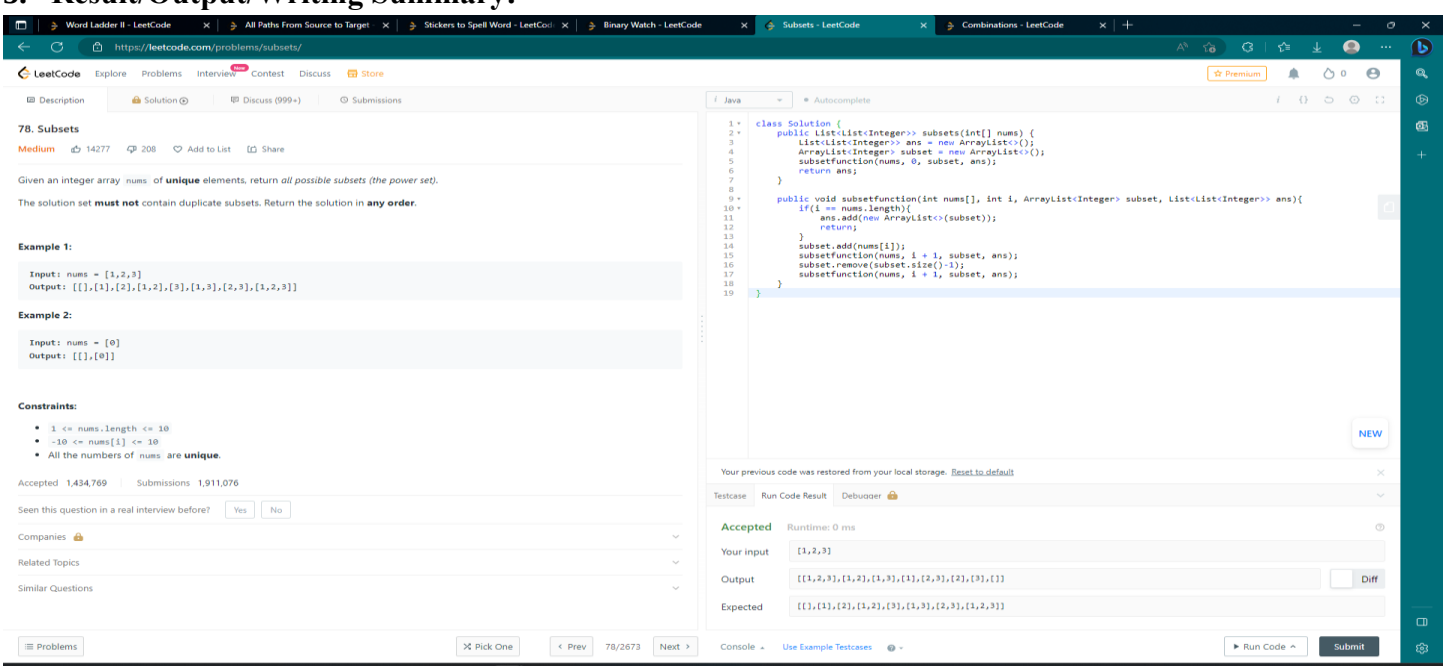


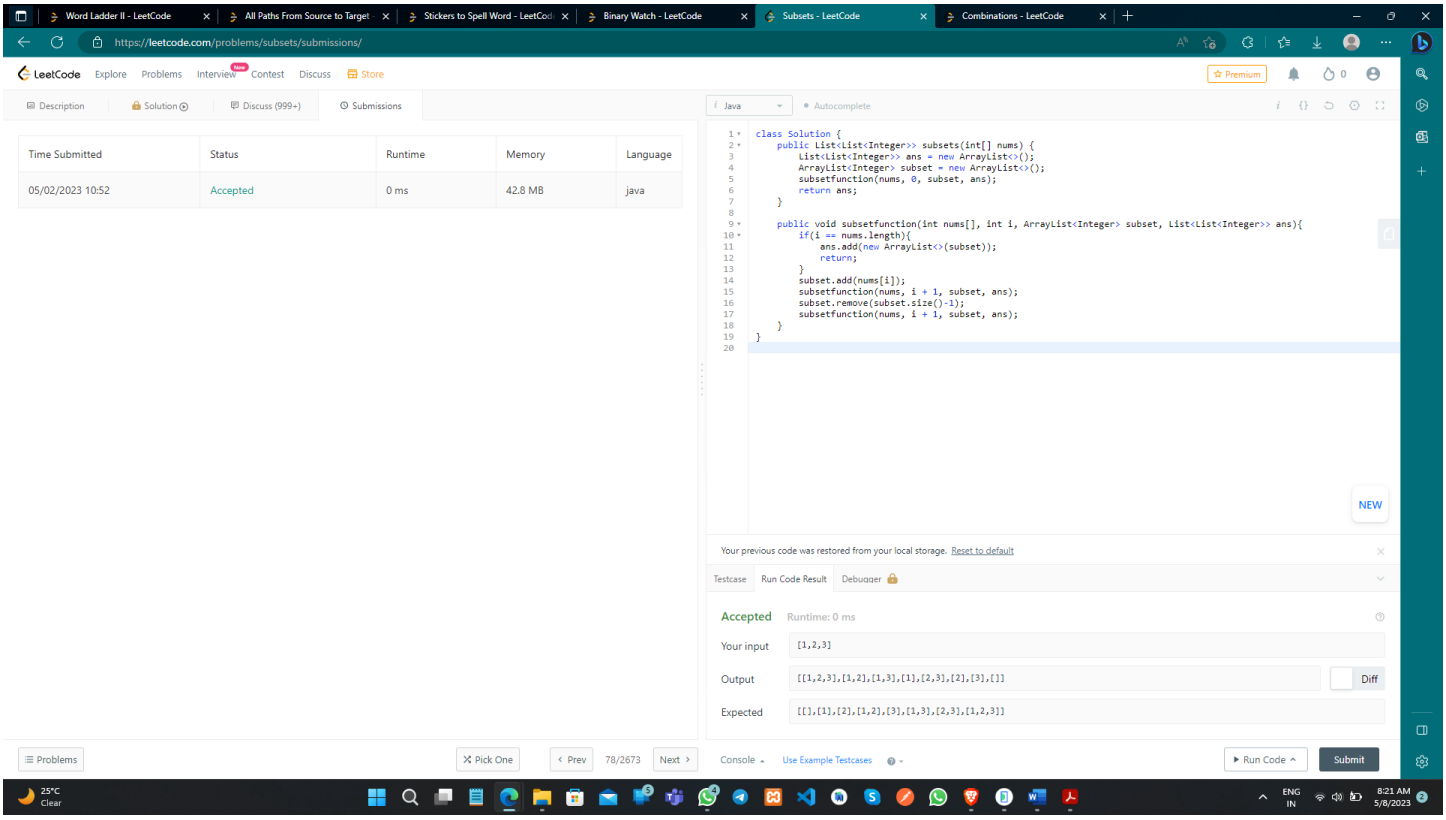


1. **Aim/Overview of the practical:**
   Q.2 Subsets.
   https://leetcode.com/problems/subsets/
2. **Apparatus / Simulator Used:**
   - Windows 7 or above
   - Google Chrome
3. **Objective:**
   - To understand the concept of Backtracking.
   - To implement the concept of Subsets.
4. **Code:**

```
class Solution {
    public List<List<Integer>> subsets(int[] nums) {
        List<List<Integer>> ans = new ArrayList<>();
        ArrayList<Integer> subset = new ArrayList<>();
        subsetfunction(nums, 0, subset, ans);
        return ans;
    }
    public void subsetfunction(int nums[], int i, ArrayList<Integer> subset, List<List<Integer>> ans){
        if(i == nums.length){
            ans.add(new ArrayList<>(subset));
            return;
        }
        subset.add(nums[i]);
        subsetfunction(nums, i + 1, subset, ans);
        subset.remove(subset.size()-1);
        subsetfunction(nums, i + 1, subset, ans);
    }
}
```

5. **Result/Output/Writing Summary:**





Learning outcomes (What I have learnt):

- Learned the concept of Backtracking Algorithm such as DFS and so on.
- Learnt about All Path from Source to Target & Subsets.