DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Experiment-9

Student Name: Kanishk Soni **UID: 20BCS9398**
Branch: BE-CSE **Section/Group:20BCS_DM-708B**
Semester: 6th **Subject Name: Competitive Coding-II**
Subject Code: 20CSP-351

AIM: To demonstrate the concept of Backtracking.

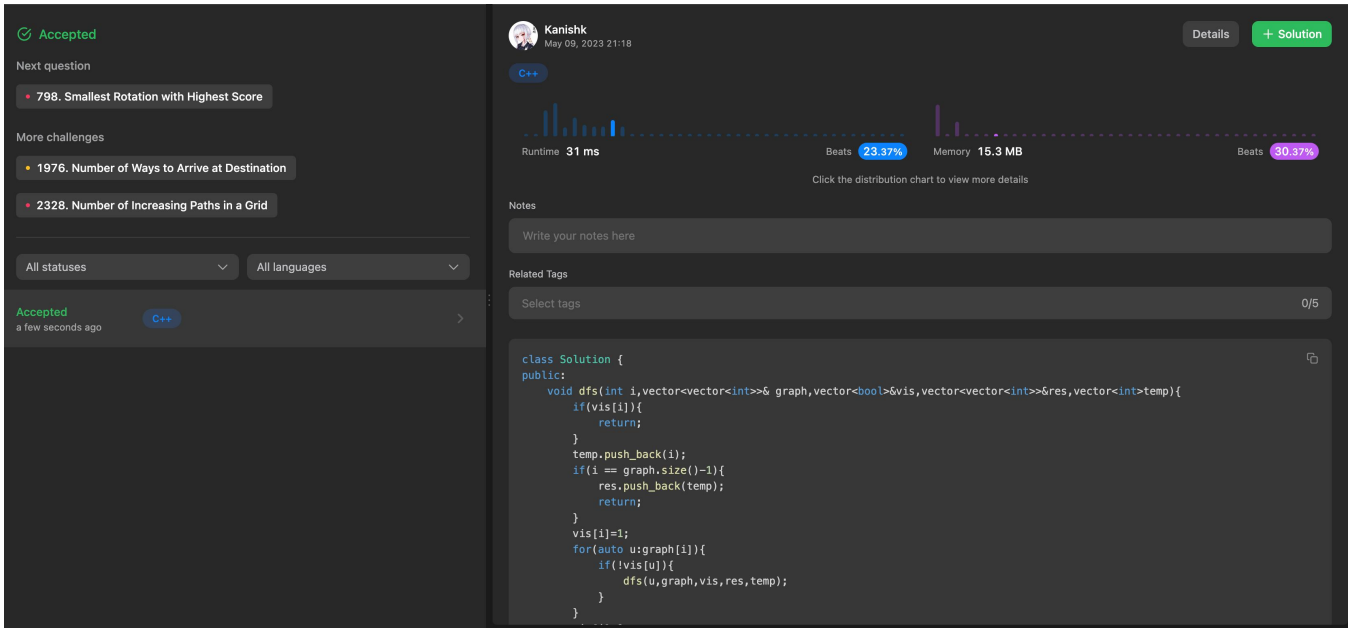
Problem1: All Paths From Source to Target
https://leetcode.com/problems/all-paths-from-source-to-target/

Program Code:

```
class Solution {
public:
    void dfs(int i,vector<vector<int>>&
             graph,vector<bool>&vis,vector<vector<int>>&res,vector<int>temp){
        if(vis[i]){
            return;
        }
        temp.push_back(i);
        if(i == graph.size()-1){
            res.push_back(temp);
            return;
        }
        vis[i]=1;
        for(auto u:graph[i]){
            if(!vis[u]){
```

```
                dfs(u,graph,vis,res,temp);
            }
        }
        vis[i]=0;
    }
    vector<vector<int>> allPathsSourceTarget(vector<vector<int>>& graph) {
        vector<bool>vis(graph.size(),0);
        vector<int>temp;
        vector<vector<int>>res;
        dfs(0,graph,vis,res,temp);
        return res;
    }
};
```

Output:



Probem2: Combinations
https://leetcode.com/problems/combinations/

Program Code:

```
class Solution {
private:
    void combine(int n, int k, vector<vector<int>> &output, vector<int> &temp, int start){
        if(temp.size() == k){
            output.push_back(temp);
            return;
        }
        for(int i=start; i<=n; i++){
            temp.push_back(i);
            combine(n, k, output, temp, i+1);
            temp.pop_back();
        }
    }
public:
    vector<vector<int>> combine(int n, int k) {
        vector<vector<int>> output;
        vector<int> temp;
        combine(n, k, output, temp, 1);
        return output;
    }
};
```

DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING

Output: