

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Experiment-9

Student Name: Kanishk Soni
Branch: BE-CSE
Semester: 6th
Subject Code: 20CSP-351

UID: 20BCS9398
Section/Group: 20BCS_DM-708B
Subject Name: Competitive Coding-II

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]){
```

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

```
        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:

Accepted

Next question

798. Smallest Rotation with Highest Score

More challenges

1976. Number of Ways to Arrive at Destination

2328. Number of Increasing Paths in a Grid

All statuses

All languages

Accepted

a few seconds ago

C++

Kanishk

May 09, 2023 21:18

Details

+ Solution

C++

Runtime 31 ms

Beats 23.37%

Memory 15.3 MB

Beats 30.37%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags

0/5

```
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);
            }
        }
    }
};
```

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Problem2: 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:

Accepted

Next question

78. Subsets

More challenges

39. Combination Sum

46. Permutations

All statuses

All languages

Accepted

a few seconds ago

C++

Kanishk

May 09, 2023 21:26

Details

+ Solution

C++

Runtime 30 ms

Beats 53.73%

Memory 9.2 MB

Beats 62.68%

Click the distribution chart to view more details

Notes

Write your notes here

Related Tags

Select tags

0/5

```
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;
    }
};
```