Date: / /
Page:

Day - 89

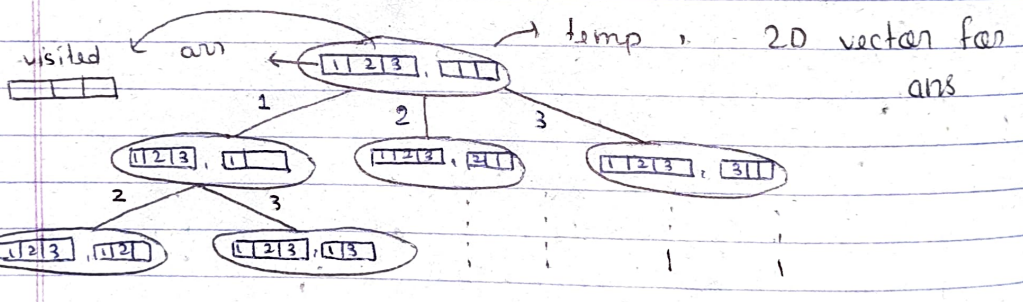
Recursion - 12

* **Permutations:**

0	1	2
1	2	3

⇒

1 2 3
1 3 2
2 3 1
2 1 3
3 1 2
3 2 1



⇒ **code permut**

```
void (int arr[], vector<vector<int>>&ans,
    vector<int>&temp, vector<bool>&visited){
        if(visited.size() == temp.size()){
            ans.push_back(temp);
            return;
        }
        for(int i=0; i<visited.size(); i++){
            if(visited[i]==0){
                visited[i]=1;
                temp.push_back(arr[i]);
                permut(arr, ans, temp, visited);
```

```
            visited[i] = 0;
            temp.pop_back();
        }
    }
```

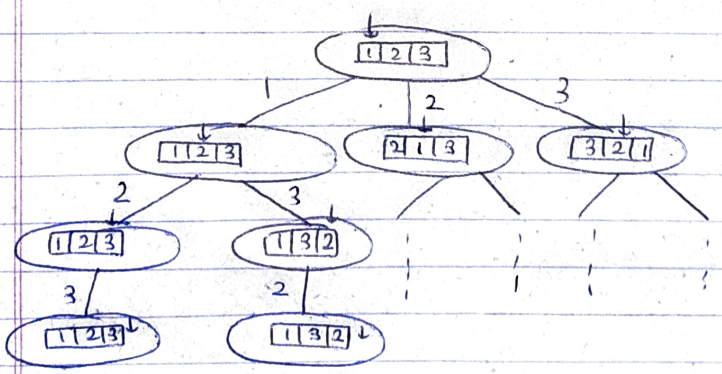
T.C → O(n * n!)
S.C → O(n)

⇒ Now, if we twist this question that you don't have to use visited & temp.

⇒ If we are not using temp & visited then we have done - all the changes in the original arr.

⇒ So, we will do that we do all the changes on that particular index first.

⇒ For this, we will swap that index value with other values.



Date: / /
Page:

```
void permut( int vector<int> &arr, vector< vector<int>
 &ans, int index){
        if (index == arr.size()){
            ans.push_back(arr);
            return;
        }
        for(int i= index ; i< arr.size() ; i++){
            swap( arr[i], arr[index ]);
            permut( arr, ans, index +1 );
            swap(arr[i], arr[index ]);
        }
}
```