DSA SERIES

- Learn Coding

Topic to be Covered today

Backtracking

LETS START TODAY'S LECTURE

What is Backtracking?

Backtracking is a refined form of recursion where:

- We **try all possible choices** one by one.
- If a choice **leads to a solution**, we continue.
- If a choice **doesn’t lead to a solution**, we **undo (backtrack)** that choice and try another.

It’s like **recursion + undo step**.

Key Concepts

When solving backtracking problems, always look for:

1. **Choice:** What options can I pick at each step?

2. **Constraint:** When should I stop exploring further?

3. **Goal/Target:** When do I know I’ve found a valid solution?

Backtracking Template

```
void backtrack(path, choices):
    if goal reached:
        save path
        return

    for each choice in choices:
        make choice
        backtrack(updated path, remaining choices)
        undo choice   // backtrack step
```

Let’s Understand the backtracking with an example :

78. Subsets

```cpp
class Solution {
public:
    vector<vector<int>> result;

    void solve(int i, vector<int>& nums, vector<int>& temp) {
        if (i >= nums.size()) {
            result.push_back(temp);
            return;
        }
        temp.push_back(nums[i]);
        solve(i + 1, nums, temp);
        temp.pop_back();
        solve(i + 1, nums, temp);
    }

    vector<vector<int>> subsets(vector<int>& nums) {

        vector<int> temp;
        solve(0, nums, temp);
        return result;
    }
};
```

More Questions to practice the backtracking

Learn coding

THANK YOU