DSA SERIES

- Learn Coding

Topic to be Covered today

Stack

LETS START TODAY'S LECTURE

Valid Parentheses (20)

```
class Solution {
public:
    bool isValid(string s) {
        stack<char> st;

        for (int i = 0; i < s.length(); i++) {
            char ch = s[i];

            if (ch == '(' || ch == '{' || ch == '[') {
                st.push(ch);
            }

            else {
                if (!st.empty()) {
                    char top = st.top();

                    if((ch == ')' && top == '(') || (ch == '}' && top == '{')||
(ch == ']’ && top == '[')){
                        st.pop();
                    }
                    else
                    {
                        return false;
                    }
                } else {
                    return false;
                }
            }
        }

        return st.empty();
    }
};
```

Maximum Nesting Depth of the Parentheses (1614)

```
class Solution {
public:
    int maxDepth(string s) {
        int currentDepth = 0 , maxDepth =0;

        for(char ch : s){
            if(ch == '('){
                currentDepth++;
                maxDepth = max(maxDepth ,
currentDepth);
            } else if(ch == ')'){
                currentDepth-- ;
            }
        }

        return maxDepth;
    }
};
```

Using Stack

```
class Solution {
public:
    int maxDepth(string s) {
        int maxDepth =0;
        stack<char> st;

        for(char ch : s){
            if(ch == '('){
                st.push(ch);
                maxDepth = max(maxDepth ,
(int)st.size());
            } else if(ch == ')'){
              st.pop();
            }
        }

        return maxDepth;
    }
};
```

Remove outermost Parentheses (1021)

```
class Solution {
public:
    string removeOuterParentheses(string s) {
        string result ="";
        int depth = 0;

        for(char ch :s){
            if(ch == '('){
                if(depth>0 )
                  result +=ch;
                depth++;
            }else if(ch == ')'){
                depth--;
                if(depth > 0) result+=ch;
            }
        }

        return result;
    }
};
```

Using Stack

```
class Solution {
public:
    string removeOuterParentheses(string s) {
        string result = "";
        stack<char> st;

        for (char ch : s) {
            if (ch == '(') {
                if (!st.empty())
                    result += ch;
                st.push(ch);
            } else {
                st.pop();
                if(!st.empty()) result +=ch;
            }
        }

            return result;
        }
    };
```

Minimum Add to make Parentheses Valid (921)

```
class Solution {
public:
    int minAddToMakeValid(string s) {
        int open = 0;
        int insertion = 0;

        for (char ch : s) {
            if (ch == '(') {
                open++;
            } else {
                if (open > 0)
                    open--;
                else {
                    insertion++;
                }
            }
        }
        return insertion + open;
    }
};
```

Using Stack

```
class Solution {
public:
    int minAddToMakeValid(string s) {
      stack<char> st;
      int insertion = 0;

      for(char ch : s){
        if(ch == '('){
            st.push(ch);
        }else{
            if(!st.empty()){
                st.pop();
            }else{
                insertion++;
            }
        }
      }

      return insertion + st.size();
    }
};
```

Learn coding

THANK YOU