Valid Parentheses

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

      // prevents seg faults
      st.push('S');

      if (s.size() <= 1) return false;

      for (i=0;i<s.size();i++){
        if (s[i] == '(' || s[i] == '{' || s[i] == '['){
          st.push(s[i]);
          stt.push(s[i]);
        }else{
          if (s[i] == ')'){
            if (st.top() != '(') return false;
            st.pop();
          }else if (s[i] == '}'){
            if (st.top() != '{') return false;
            st.pop();
          }else if (s[i] == ']'){
            if (st.top() != '[') return false;
            st.pop();
          }else{
            return false;
          }
        }
      }

      if ((stt.size())*2 != s.size()) return false;
```

```
      return true;

    }
};
```

- working solution
  - starts at beginning of string and checks if its opening parentheses
  - if it is it adds it to the stack and moves on too the next
  - if the next is not an opening paren then it checks to see if it matches the current open paren that is supposed to be closed
- Big O
  - O(n)