

2

1

4

3

10

☐ Regular Expressions in JavaScript☐ `str.search(pattern)` returns position of match☐ pattern is `/string/igm`, trailing characters are modifiers☐ CCC, strings have combinatorial expressive power. Very rich. Bootable.☐ parsing natural language with machine learning☐ bioinformatics is housed in information science☐ `c` + 1 or more instances of `c`  
`c*` 0 or more instances of `c`  
`c?` 0 or 1 instances☐ the subset of all strings which are valid JavaScript programs☐ the overlapping regular expression problem is significant lexically!☐ The parentheses problem:For a given expression involving multiple parentheses,  
determine if they are balanced.

This is an interesting example of a 'halting problem'.

A regular expression solution:

```
<button onclick="let e = '(((1+2)))'; countParens(e)">
Check Expression
</button>
```

```
<p id="report"></p>
```

```
<script>
function countParens(expr)
{
  let leftCount = expr.match(/\(/g).length;
  let riteCount = expr.match(/\)/g).length;
  let loc = document.getElementById("report");

  if(leftCount === riteCount)
    loc.innerHTML = 'parens match';
  else
    loc.innerHTML = "parens don't match";
}
</script>
```