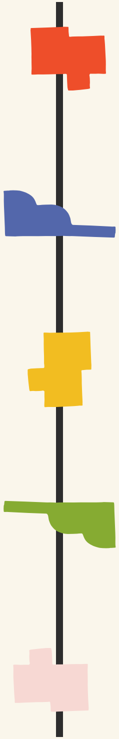
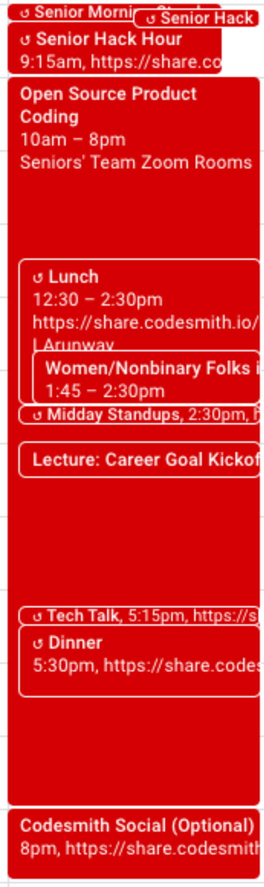
HH SOLUTION

generate-parentheses

Senior Schedule





↺ Senior Morni

↺ Senior Hack

↺ Senior Hack Hour
9:15am, https://share.co

Open Source Product Coding
10am – 8pm
Seniors' Team Zoom Rooms

↺ Lunch
12:30 – 2:30pm
https://share.codesmith.io/
l Arunway

Women/Nonbinary Folks i
1:45 – 2:30pm

↺ Midday Standups, 2:30pm, F

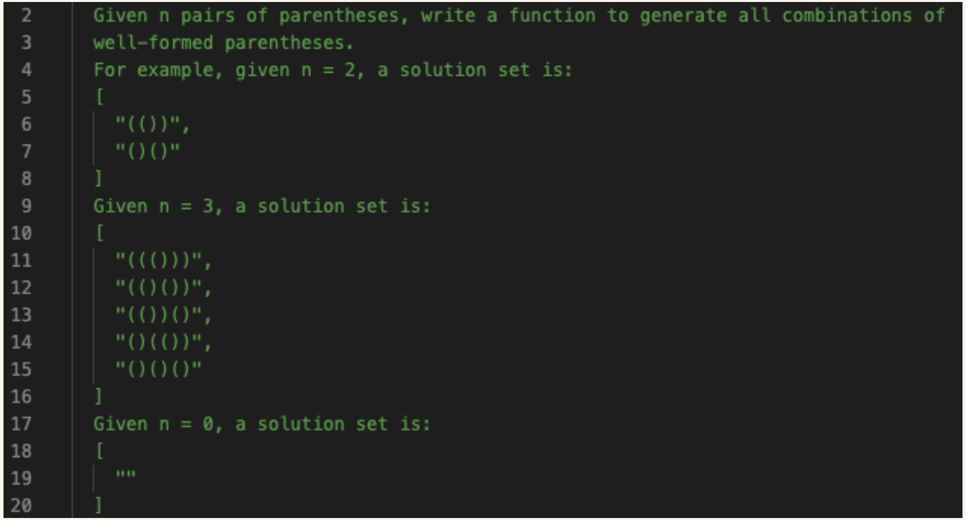
Lecture: Career Goal Kickof

↺ Tech Talk, 5:15pm, https://s

↺ Dinner
5:30pm, https://share.codes

Codesmith Social (Optional)
8pm, https://share.codesmitl

generate-parentheses



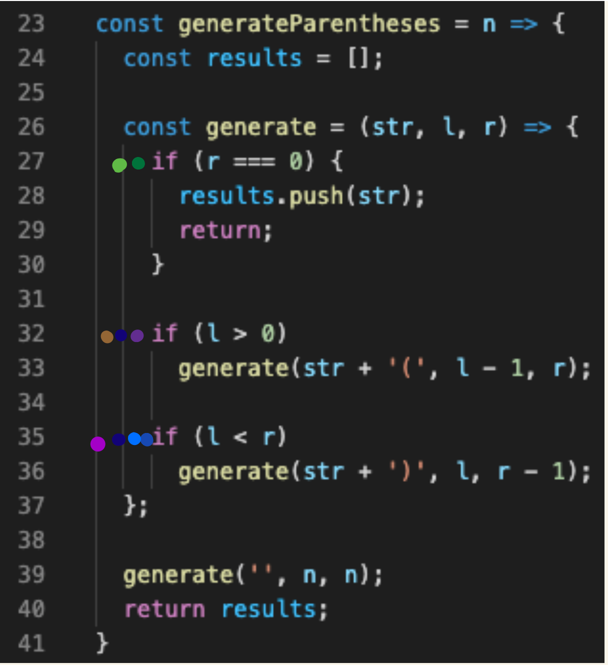
```
Given n pairs of parentheses, write a function to generate all combinations of
well-formed parentheses.
For example, given n = 2, a solution set is:
[
  "(())",
  "()()"
]
Given n = 3, a solution set is:
[
  "((()))",
  "(()())",
  "(())()",
  "()(())",
  "()()()"
]
Given n = 0, a solution set is:
[
  ""
]
```

strategy:

recursion

input: number

output: array of strs

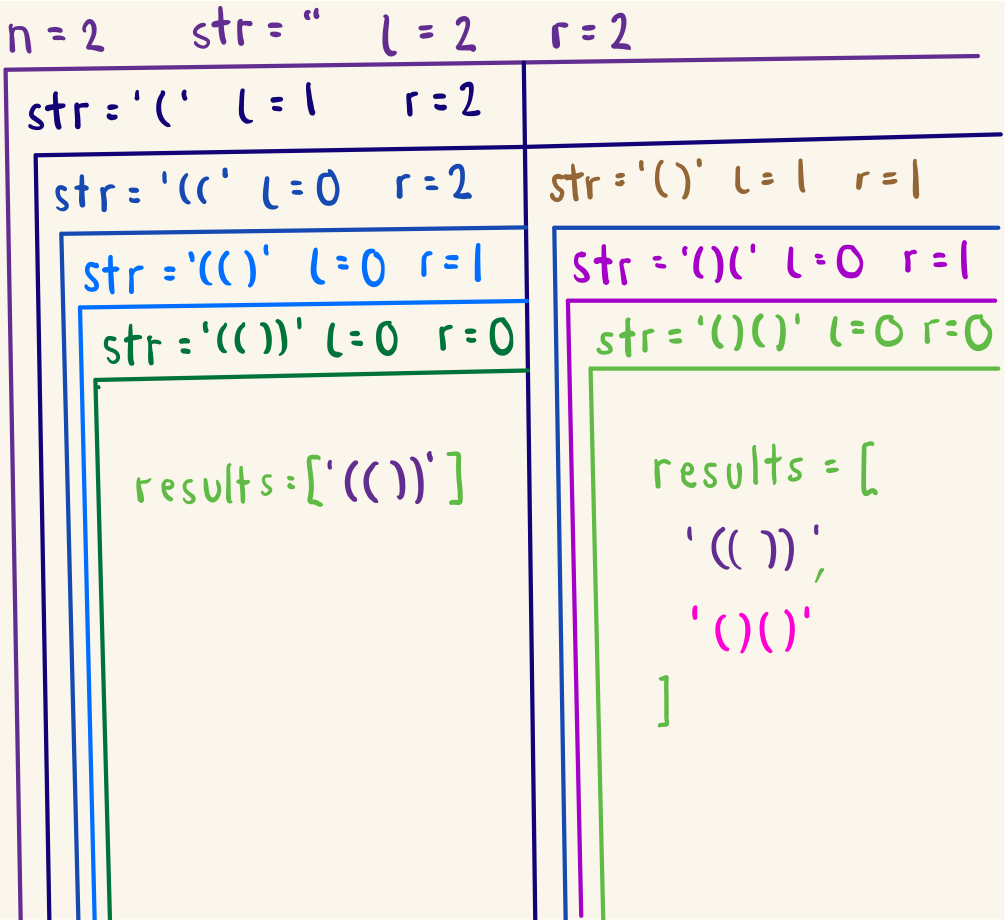


```
const generateParentheses = n => {
  const results = [];

  const generate = (str, l, r) => {
    if (r === 0) {
      results.push(str);
      return;
    }

    if (l > 0)
      generate(str + '(', l - 1, r);

    if (l < r)
      generate(str + ')', l, r - 1);
  };

  generate('', n, n);
  return results;
}
```

generateParens(2);

//expect

['(())' , '()()']



n = 2 str = '' l = 2 r = 2

str = '(' l = 1 r = 2

str = '((' l = 0 r = 2

str = '(()' l = 0 r = 1

str = '(())' l = 0 r = 0

results = ['(())']

str = '()' l = 1 r = 1

str = '()(' l = 0 r = 1

str = '()()' l = 0 r = 0

results = [
'(())',
'()()'
]

HH CHALLENGE

get-all-products