Leetcode 22: Generate parentheses.

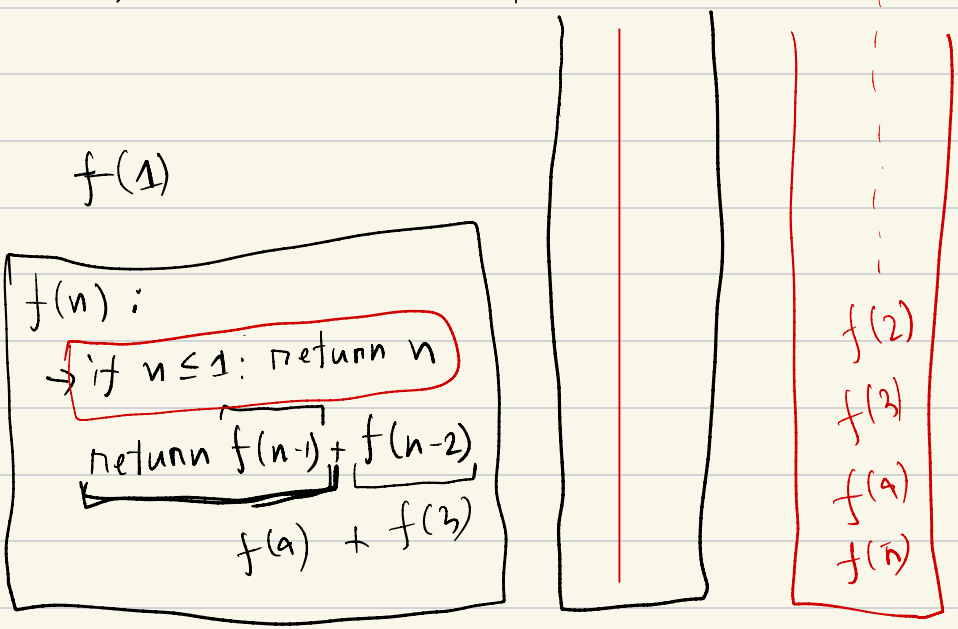
Given n, output all valid parentheses of length 2n.

S += "("

left ++
right ++

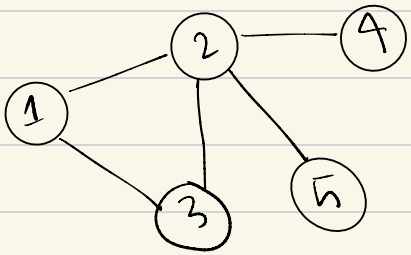
$n \geq left \geq right$

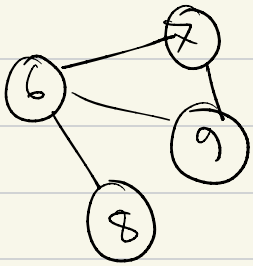
$f(n) = f(n-1) + f(n-2)$ | $f(0) = 0$, $f(1) = 1$



f(n)

DFS:





n = 3

((())) (()())

```java
private void pattern(List<String> v, String s, int left, int right, int n) {
    if (left == right && left == n) {
        v.add(s);
        return;
    }

    if (right > left || left > n) {
        return;
    }

    if (left >= right && left < n) {
        pattern(v, s + '(', left + 1, right, n);
    }

    if (left > right) {
        pattern(v, s + ')', left, right + 1, n);
    }
}
```

n = 3

S =

(()())

((

((, 2, 0

(, 1, 0

, 0, 0

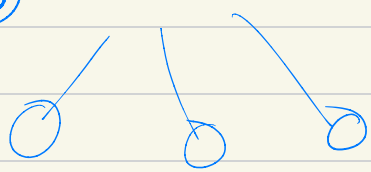
left 2

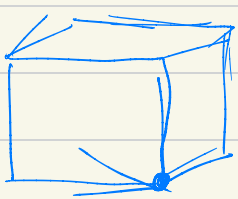
right 0

$$f(n) = 2f(n-1) + n f(n-2) + 5n^2 f(n-3)$$

$5^3 = 125$

$n = 5$





$\mathbb{R}^n$

$\{(0,0,0)$

---

n – Given

you have to tell me how many binary numbers can you make with n bits (0/1).

$\boxed{2}\ \boxed{2}\ \cdots\ \boxed{2} \rightarrow 2^n$

2 0's cannot be together.

0 1 1 1 0 1 0 1

0 0 1 1 1 0 0 1 1 1

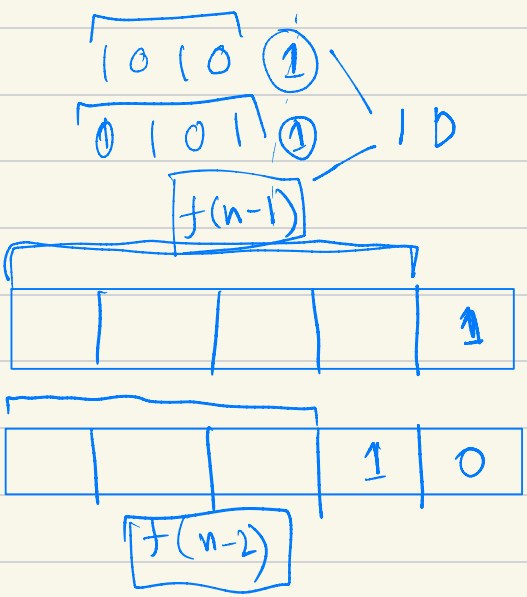
1 1 0 1 1 0 1 0 1

(K) 0 1 0 1 0 1 0 1 0

1 0 1 0 1

0 1 0 1 1

1 0



S_1 → all valid strings that ends with 1

S_2 → all valid strings ends with 0.

$f(n)$ = ans.

$$f(n) = f(n-1) + f(n-2)$$

$f(0) = 0$

$f(1) = 2$