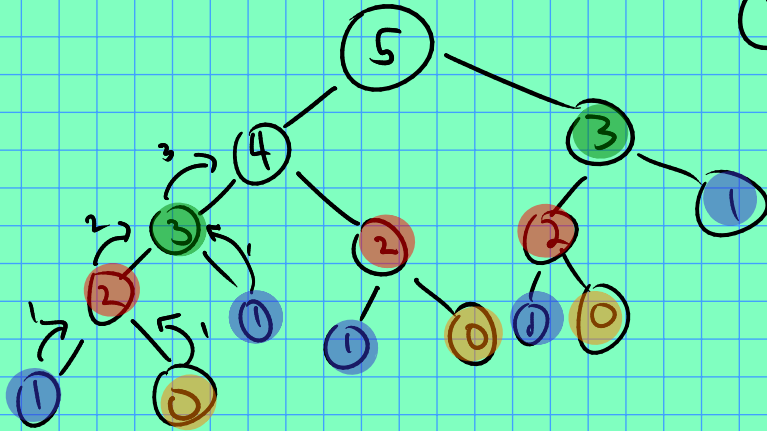


Fibonacci revisited:  $out \rightarrow (1, 1, 2, 3, 5, 8, \dots)$   
 $n \rightarrow \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & \dots \\ & \swarrow & \searrow & & & & \end{matrix}$

```
int f(int n)
{
    if (n < 2) return 1;
    return f(n-1) + f(n-2);
}
```

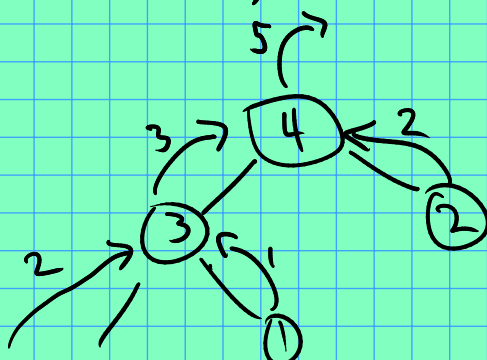
// what happens when we call  $f(5)$ ?

$\textcircled{x} \equiv \text{calling } f(x)$



Memorization:  $(M \equiv \text{lookup table of } (n, f(n)) \text{ pairs})$

```
int f(int n, map<int, int> & M)
{
    if (n < 2) return 1; // base case
    if (M.find(n) != M.end()) {
        // have I already computed f(n)?
        return M[n];
    }
    int r = f(n-1, M) + f(n-2, M);
    M[n] = r;
    return r;
}
```



$M = \{ (2, 2), (3, 3), (4, 5) \}$

