

Fibonacci sequence w/ recursion.

Let's try it recursively:

base case \rightarrow

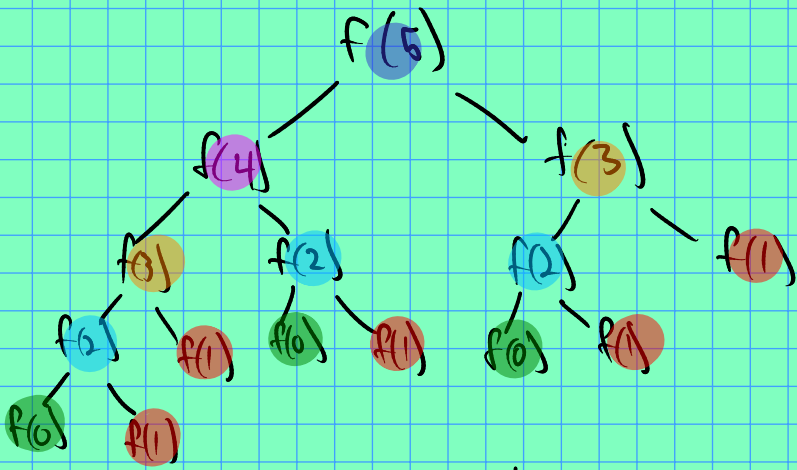
1	1
0	1

 \leftarrow outputs

\leftarrow inputs

```
int fib(int n) {  
    if (n < 2) return 1; // base case.  
    return fib(n-1) + fib(n-2);  
}
```

Let's draw the recursion tree for $n=5$:



$\approx 2^n$ steps!

How to improve, yet still leverage the expressiveness of recursion?

Maybe we can make it less forgetful...

Technique: "memoization".

Idea: use an array (or vector, or map...) to record answers in a database. Then check database before making any recursive calls.

```
map<int, int> answers;
```

```
int fib(int n) {  
    if(n < 2) return 1;  
    // check for answer in DB:  
    if(answers.find(n) != answers.end())  
        return answers[n];  
    // n not in DB. Do actual work...  
    int f = fib(n-1) + fib(n-2);  
    // add answer to DB:  
    answers[n] = f;  
    return f;  
}
```