30 Sept. 2022

Fibonacci Numbers

A sequence of numbers $f_0, f_1, f_2, \ldots$ defined as follows:

$$f_i = \begin{cases} 0, & i = 0 \\ 1, & i = 1 \\ f_{i-1} + f_{i-2}, & i \geq 2 \end{cases}$$

Question: What is f_8?

INPUT: int i (non-negative)
OUTPUT: f_i (an integer)

```
FIB (i)
1: if i ≤ 1
2: | return i
3: endif
4: return FIB(i-1) + FIB(i-2)
```

What is the runtime?

$T(n)$ = runtime of FIB(n)

$T(n) = \Theta(1) + T(n-1) + T(n-2)$

$= 1.618\ldots^n$

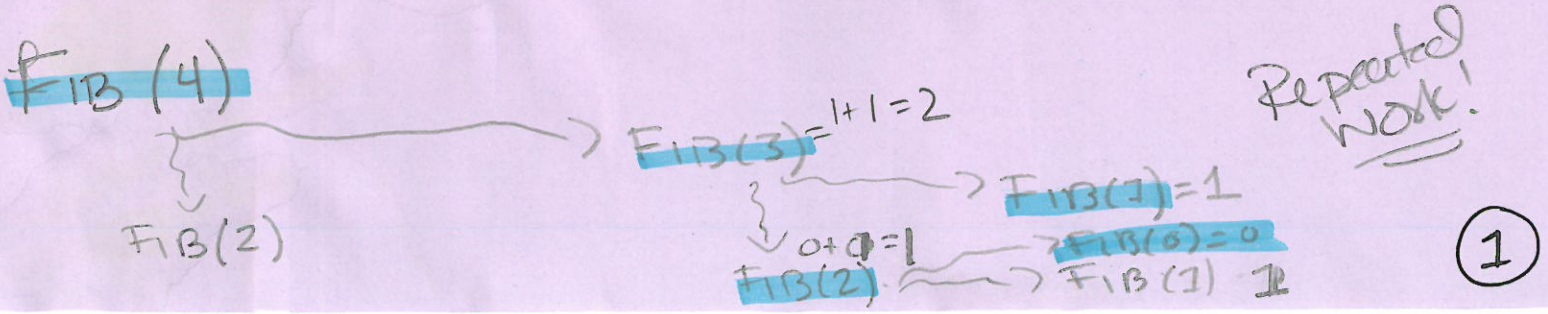
$= \phi^n$

$T(n) \leq \Theta(1) + T(n-1) + T(n-1)$

Similar recurrence:

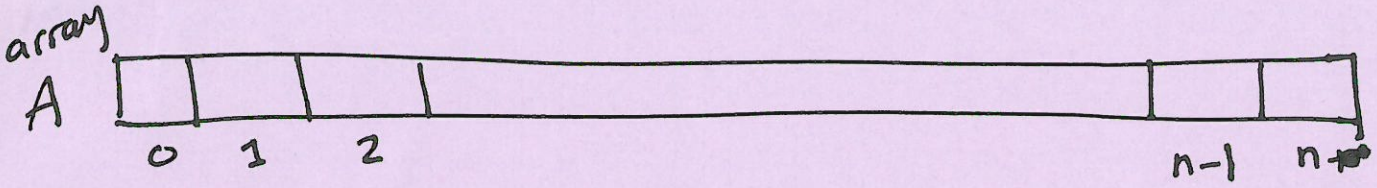
$T_1(n) = 2T_1(n-1) + \Theta(1)$

$= 2^n$



Q: How many sub-problems are there actually? (distinct problems, for FIB(n))

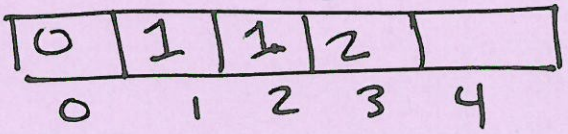
There are only $n+1$ problems!



A[i] stores the sol'n to my ith problem

Take 1: Memoization

Take 2: Think carefully about the order of computation



Data Structures for storing n values:

① Hash Map

- maps keys to bins.
  (integers) (integers)

  e.g., Map 1: $i \longmapsto i \bmod 5$

  Map 2: $i \longmapsto 6$

  These are both "valid" hashes, ..but... there are typically properties that you would want your has to have, e.g., collision resistent.

- In a "good" hash:
  - → each bin has $\Theta(1)$ values, in expectation.
  - → possible to have $\Theta(n)$ values!
  - → to find item i, look in $h(i)$ bin ... takes
    - $\Theta(1)$ in Expectation
    - $\Theta(n)$ in Worst-case
  - → to add a new item to hash map with input key i
    - $\Theta(1)$ in Expectation
    - Worst-case: depends on the data structure of the bin.

② Array

→ Find is $\Theta(1)$

→ Append / Add a new item: $\Theta(n)$

→ update a value. $\Theta(1)$

③ List / Linked List

→ Add new item: $\Theta(1)$

→ Find is $\Theta(n)$

→ update is $\Theta(n)$