Michael Cooper 17C Midterm - Problem 6

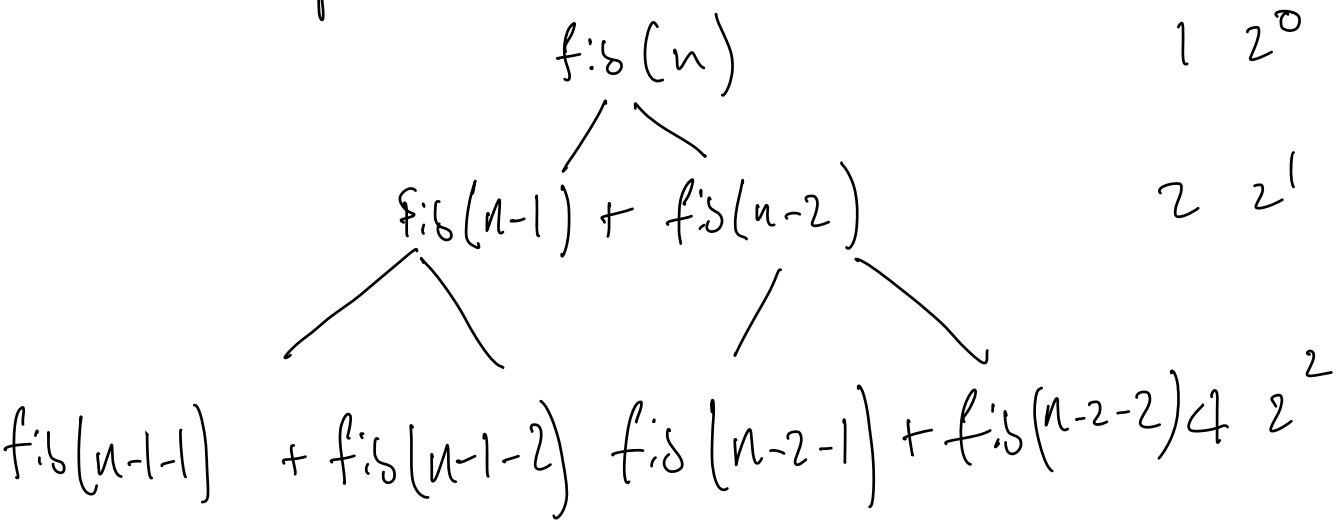
Recursive: $fib(n) = \begin{cases} 0 & n \leq 0 \\ 1 & n = 1 \\ fib(n-1) + fib(n-2) & \text{otherwise} \end{cases}$

- fib(n) is $O(2^n)$ because fib(0) is $O(1)$, Fib(1) is $O(1)$ and fib(n) is $O(2^n)$.

  Since $O(1) + O(1) + O(2^n) = O(\max(1, 1, 2^n)) = O(2^n)$

- fib(n) is $O(2^n)$ because each call for Fib(n) produces 2 calls for fib

Example:



Non Recursive:

```
fib(n) {
  ...
  for(i=2, i<n; i++){
    ...
  }
}
```

fib is $O(n)$ because when $n \geq 2$

$$\sum_{i=2}^{n} fib_i + fib_{i+1}$$