Hands-On 4

0) 1) fibonacciopy in link folder

2) Each fib() call results in 2 additional calls until n is = 1,0. This indicates an exponential growth

$T(n) = T(n-1) + T(n-2) + O(1)$

Base: $T(1) = T(0) + 1$ $\quad T(0) = T(1) = 0$

$T(2) = T(1) + T(0) + 1$

$= 1 + 0 + 1 = 2$

Assuming $T(n-1) \approx T(n-2)$

Backwards Substitution

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

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

$= 4T(n-2) + 3 \quad T(n-2) = 2T(n-3)+1$

$= 4(2T(n-3)+1)+3$

$= 8T(n-3)+7 \quad T(n-3) = 2T(n-4)+1$

$= 8(2(n-4)+1)+7$

$= 2(2(2(2T(n-4)+1)+1)+1)+1)$

$= 16T(n-4)+15$

Pattern for K substitutions

$2^K T(n-k) + (2^K - 1)$ $\quad$ K builds until K = n

$T(n) = 2^n T(0) + (2^n - 1) = 2^n + 2^n - 1 = O(2^n)$

exponential

3) Improvements:

- Store numbers in [illegible]
- Dynamic Programming [illegible]