what you should be able to do for divide + conquer:

- decide whether a divide + conquer alg is correct
- be able to trace through the execution of a D+C alg
- know ~~mergesort~~, split multiply, and Karatsuba's alg pseudo code
- given a rec. alg., write the recurrence for its run time
- given a recurrence, be able to draw recursion tree, sum work at levels, generalize, know depth, figure out overall runtime

mergesort:
$T(n) = 2T(\frac{n}{2}) + \Theta(n)$,
$T(1) = \Theta(1)$
$(T(1) = 1)$

- write divide + conquer algs
- prove that divide + conquer algs are correct by induction.

Dynamic Programming

Fibonacci numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

F_0 F_1 F_2 F_3

$$F_n = \begin{cases} 0 & \text{if } n=0 \\ 1 & \text{if } n=1 \\ F_{n-1}+F_{n-2} & \text{if } n>1 \end{cases}$$

base cases

recursive case

BONUS!!

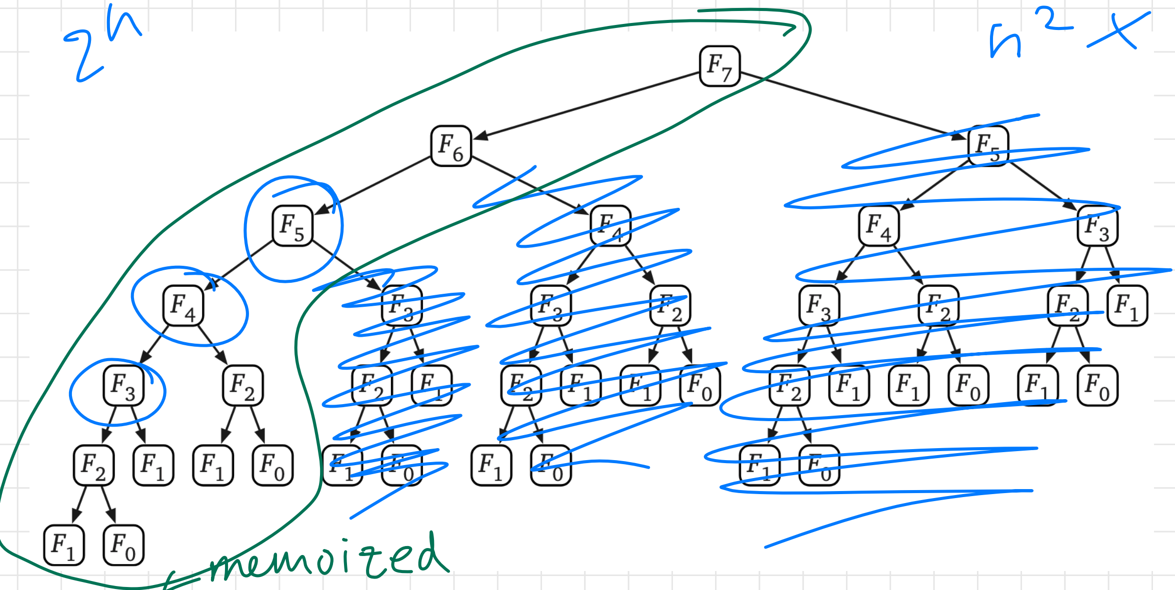
what is the recurrence for the fibo runtime?

```
fibo(n):
  x if n=0: return 0
  x if n=1: return 1
  if n>1: return fibo(n-1) + fibo(n-2)
```

recursive calls   non-rec. work

$$T(n) = \overbrace{T(n-1)+T(n-2)}^{\text{recursive calls}} + \overbrace{1}^{\text{non-rec. work}}$$

$T(0)=1$, $T(1)=1$




```
MemFibo(n):
    if n = 0 : return 0
    if n = 1 : return 1
    else if n > 1 :
        if F[n] is unfilled :
            F[n] = MemFibo(n-1) + MemFibo(n-2)
        return F[n]
```