Recursion 2

- power a^n
  - method 1
  - method 2
  - method 3

- pow(a, n, p_m)

break? ←

- TC & SC of recursive codes
  - sum, pow, fact, ...
  - Fib

no overflow

$a \geq 0$ $n \geq 0$

P1 Given a, n find a^n using recursion

ex

a	n	a^n
2	5	$2^5 = 32$
3	4	$9 \times 9 = 81$

✓ ① Assumption
✓ ② Main Logic
✓ ③ Base Condition

$n==0 \rightarrow a^n = 1$

pow1

```
int pow1(int a, int n){
  if(n==0) ret 1
  ret pow1(a,n-1)xa
}
```

$3^4 = 3 \times 3 \times 3 \times 3$

3×3^3 → subproblem

$a^n = \underbrace{a \times a \times a \times \cdots \times a}_{n \text{ times}}$

$= a \times a^{n-1}$

$a^n = \underbrace{a \times a \times a \times \cdots \times a}_{n \text{ times}}$

pow2

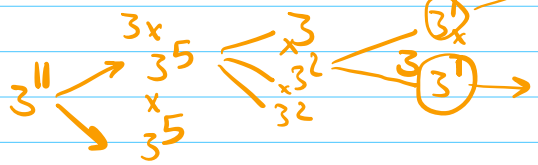
```
int pow2(int a, int n){
  if(n==0) ret 1
  if(n%2==0){ even?
    ret pow2(a, n/2)*
        pow2(a, n/2)
  }else{ odd?
    ret pow2(a, n/2)*
        pow2(a, n/2) *a
  }
}
```

$3^4 = 3 \times 3^3 = 3^2 \times 3^2$

if n is even $a^n = a^{n/2} \times a^{n/2}$

$3^7 = 3 \times 3^6 = 3 \times 3^3 \times 3^3$

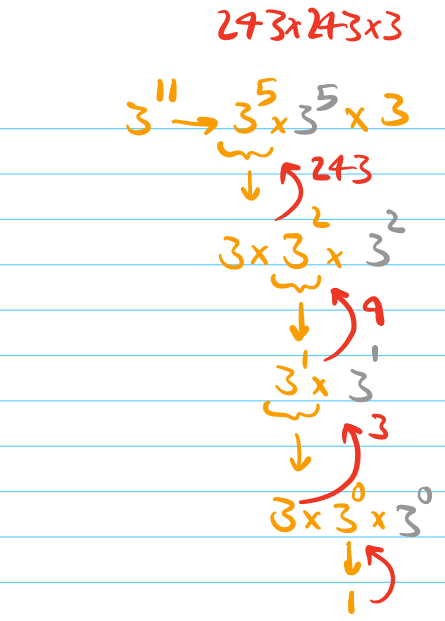
if n is odd

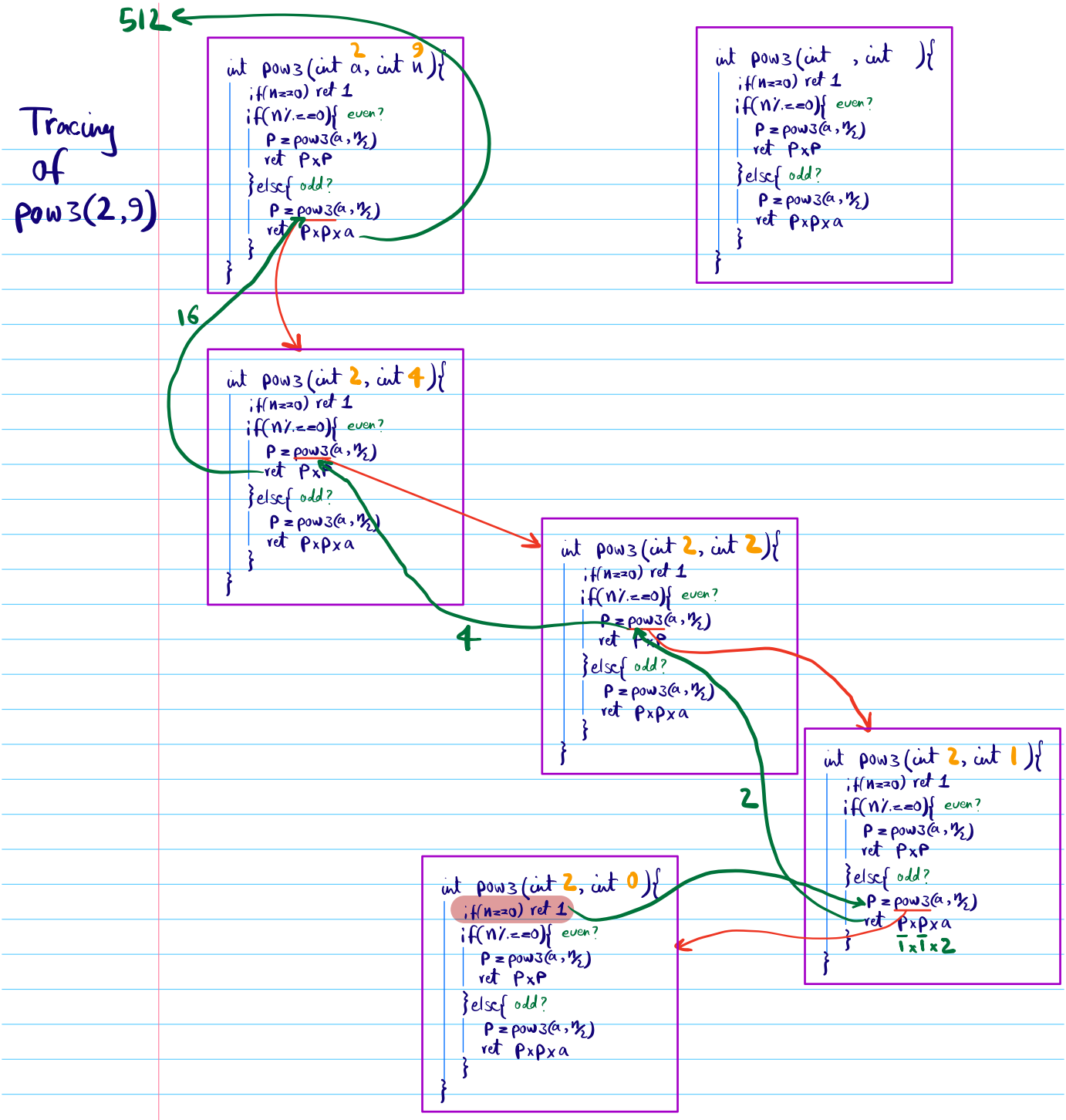


* fast power

pow3

```
int pow3(int a, int n){
    if(n==0) ret 1
    if(n%2==0){ even?
        p = pow3(a, n/2)
        ret p*p
    }else{ odd?
        p = pow3(a, n/2)
        ret p*p*a
    }
}
```




Tracing of pow3(2,9)
512
16
4
2
1
int pow3(int a, int n){
if(n==0) ret 1
if(n%2==0){ even?
P = pow3(a, n/2)
ret PxP
}else{ odd?
P = pow3(a, n/2)
ret PxPxa
}
}
int pow3(int 2, int 9){
int pow3(int 2, int 4){
int pow3(int 2, int 2){
int pow3(int 2, int 1){
int pow3(int 2, int 0){
if(n==0) ret 1
1x1x2

P2 Given a, n, m, calculate $a^n \,\%\, m$

$1 < a <= 10^9$
$1 < n <= 10^9$
$2 < m <= 10^9$

2^{10000} overflow?

$a^n \,\%\, m = [\overbrace{a \times a \times a \cdots \times a}^{n \text{ times}}] \,\%\, m$

$(a * b) \,\%\, P = [(a \,\%\, P) * (b \,\%\, P)] \,\%\, P$

$= [\underbrace{a \,\%\, m \times a \,\%\, m \cdots}_{n/2} \times \underbrace{a \,\%\, m}_{n}] \,\%\, m$ $\times(a)$

TC? why?

```
int powMod(int a, int n, int m){
    if(n==0) ret 1
    if(n%2==0){ even?
        long P = powMod(a, n/2, m)
        ret P*P %m  ---> (P%m)*(P%m)%m
    }else{ odd?
        long P = powMod(a, n/2, m)
        ret ((P*P%m)*a%m) %m
    }
}
```

break?

TC for recursive code using recursive relation

ex: sum

```
int sum(n) {
    if(n==1) ret 1 → O(1)
    ret sum(n-1)+n → O(1)
              T(n-1)
}
```

$T(N) = T(N-1) + 1$

step	
1	$T(n) = T(n-1) + 1$
2	$= T(n-2) + 1 + 1$
3	$= T(n-3) + 1 + 1 + 1$
	⋮
n-1	$= T(n-(n-1)) + n-1$
	$= T(1) + (n-1)$

$T(1) = 1$

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

TC: $O(n)$

after k steps $\quad k = (n-1)$

$T(n) = T(n-k) + k$

$n - k = 1$

$\Rightarrow k = n-1$

ex: fact

```
int fact(n){
    if(n==1) ret 1
    ret  n*fact(n-1)
}
```

Quiz

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

TC: $O(n)$

pow1

```
int pow1(int a, int n){
  if(n==0) ret 1      O(1)
  ret pow1(a, n-1)xa  O(1)
}
```

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

TC: $O(n)$

pow2

```
int pow2(int a, int n)
  if(n==0) ret 1      O(1)
  if(n%2==0){         O(1)
    ret pow2(a, n/2) x
        pow2(a, n/2)
  }else{ //odd
    ret pow2(a, n/2)x
        pow2(a, n/2)xa
  }
}
```

$T(N) = 2T(\frac{N}{2}) + 1$

$T(n) = T(\frac{n}{2}) + T(\frac{n}{2}) + 1$

step	
①	$= 2T(\frac{n}{2}) + 1$
②	$= 2 \times [2T(\frac{n}{4}) + 1] + 1 \leftarrow 4T(\frac{n}{4}) + 3$
③	$= 4[2T(\frac{n}{8}) + 1] + 3 \leftarrow 8T(\frac{n}{8}) + 4 + 3$
④	$= 8[2T(\frac{n}{16}) + 1] + 7 \leftarrow 16T(\frac{n}{16}) + 8 + 7$
⑤	$= 16[2T(\frac{N}{32}) + 1] + 15$

$T(0) = 1$

$T(1) = 1$ or 2

$\frac{n}{2^k} = 1$ → assum n is the closest larger power of 2 as an upper bound

after k steps

$$T(n) = \underbrace{2^k}_{n} \times T\left(\underbrace{\frac{n}{2^k}}_{n}\right) + \underbrace{(2^k - 1)}_{n} = n \times \underbrace{T(1)}_{1 \text{ or } 2} + (n-1)$$

$\frac{n}{2^k} = 1 \Rightarrow n = 2^k$

$k = \log_2 n$

$\Rightarrow T(n) = 3n - 1$

$TC = O(n)$

pow3

Quiz

```
int pow3(int a, int n){
    if(n==0) ret 1
    int p = pow3(a, n/2)
    if(n%2==0){
      ret p x p
    }else{//odd
      ret p x p x a
    }
}
```

$T(0)=1$

$T(1)=1$ or 2

Steps

① $T(n)=T(\frac{n}{2})+1$

② $=T(\frac{n}{4})+\overbrace{1+1}^{2}$

③ $=T(\frac{n}{8})+\overbrace{1+1+1}^{3}$

④ $=T(\frac{n}{16})+\overbrace{1+1+1+1}^{4}$

⋮

k

after k step

$T(n)=T(\frac{n}{2^k})+k = \overbrace{T(1)}^{2}+\log_2 n$

$\frac{n}{2^k}=1 \Rightarrow 2^k=n \Rightarrow k=\log_2 n$

$T(n)=2+\log_2 n \longrightarrow TC: O(\log_2 n)$

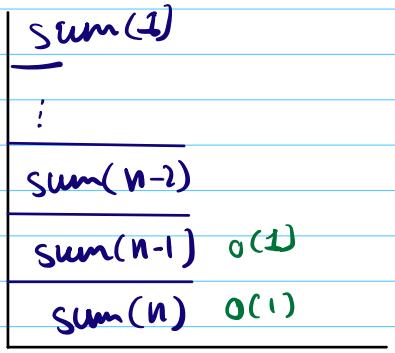
pow mod $O(\log_2 n)$

Space Complexity for recursion?

sum

```
int sum(n) {
    if(n==1) ret 1
    ret sum(n-1)+n
}
```

SC: O(n)



fact

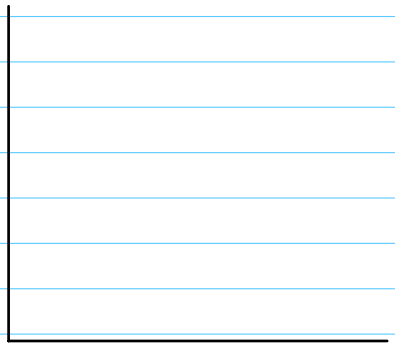
```
int fact(n){
    if(n==1) ret 1
    ret  n*fact(n-1)
}
```

SC: O(n)

pow1

```
int pow1(int a, int n){
    if(n==0) ret 1
    ret pow1(a, n-1)xa
}
```

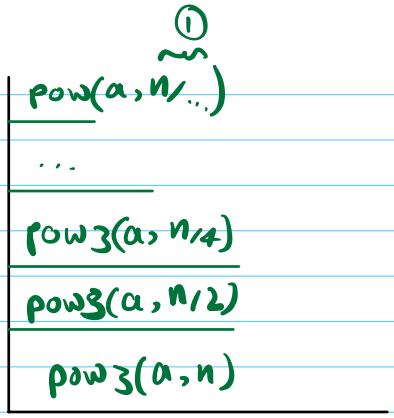
SC: O(n)



Pow3

```
int pow3(int a, int n){
  if(n==0) ret 1
  int p = pow3(a, n/2)
  if(n%2==0){
    ret p x p
  }else{//odd
    ret p x p x a
  }
}
```

SC: $O(\log_2 n)$



TC, SC

Fib

$F(n) = F(n-1) + F(n-2)$ $\quad F(0)=0$, $F(1)=1$

```
int Fib(n){
  if(n==0) ret 0
  if(n==1) ret 1
  ret Fib(n-1)+Fib(n-2)
}
```

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

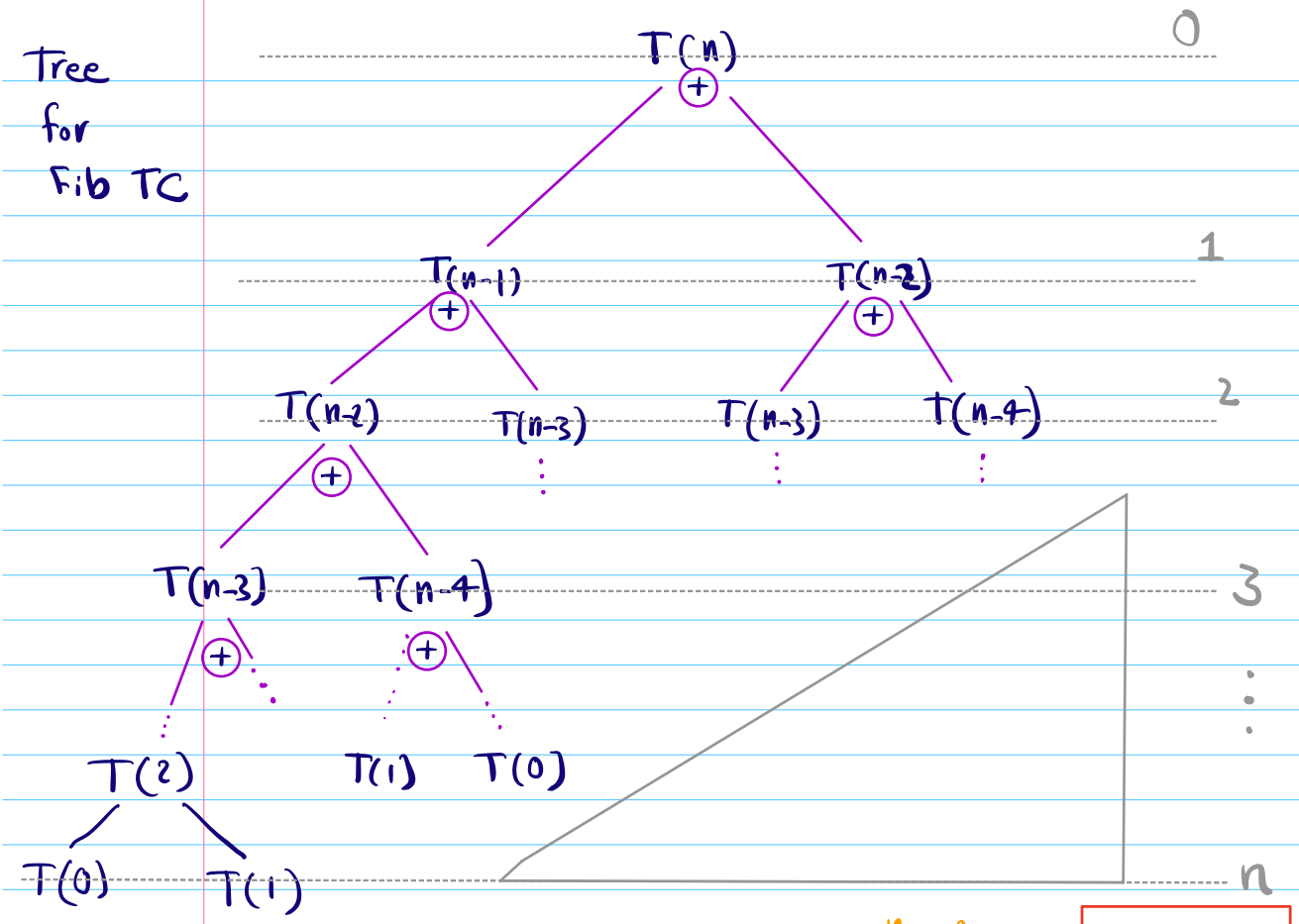
$= T(n-2)+T(n-3)+T(n-2)+1+1$

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

$= 2T(n-2)+T(n-4)+T(n-5)+3$

$T(n)=$ expansion will be cumbersome

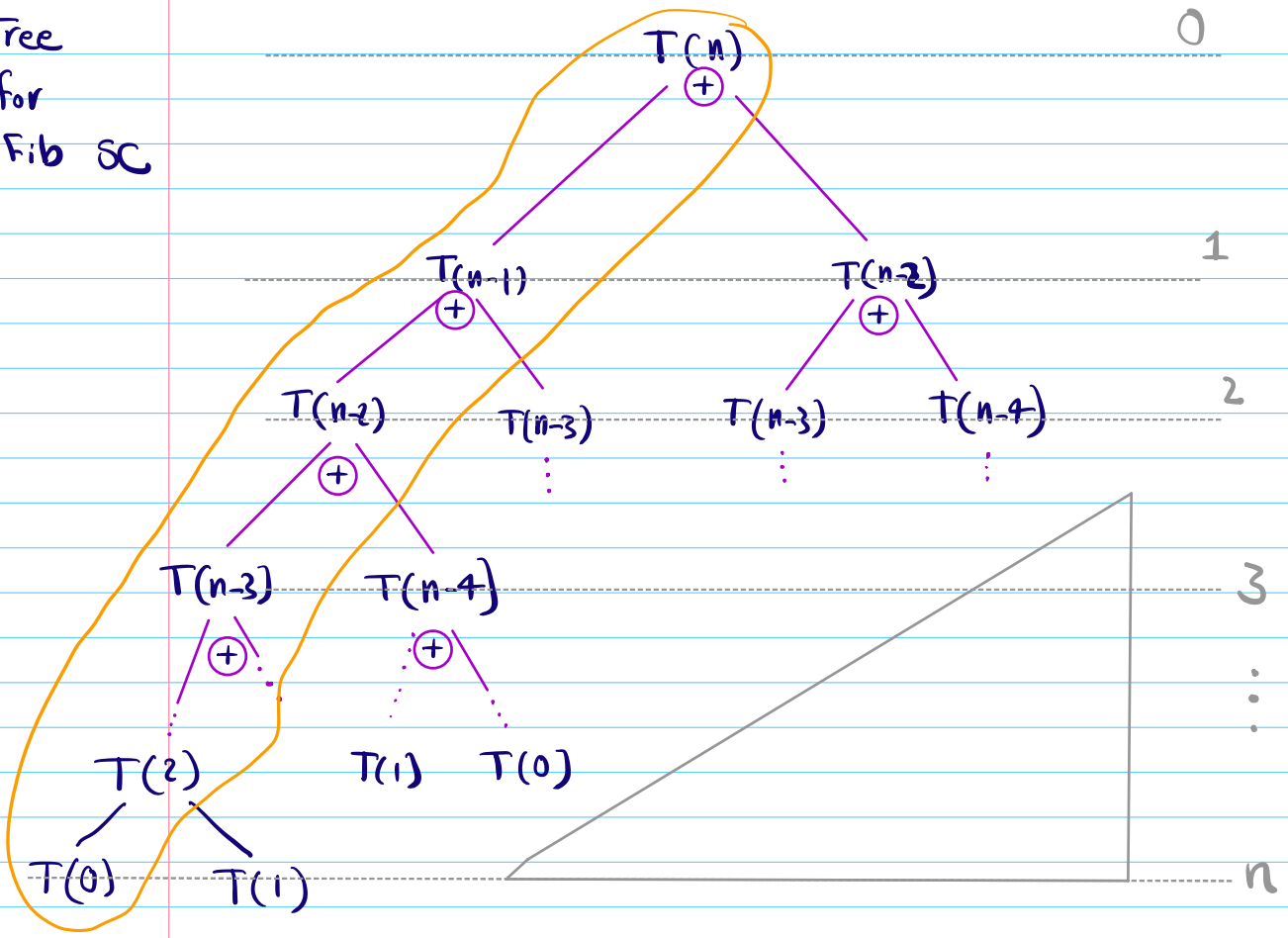
Tree for Fib TC



$$2^0 + 2^1 + 2^2 + 2^3 + \cdots + 2^n = \frac{1(2^n - 1)}{2 - 1} \sim O(2^n)$$

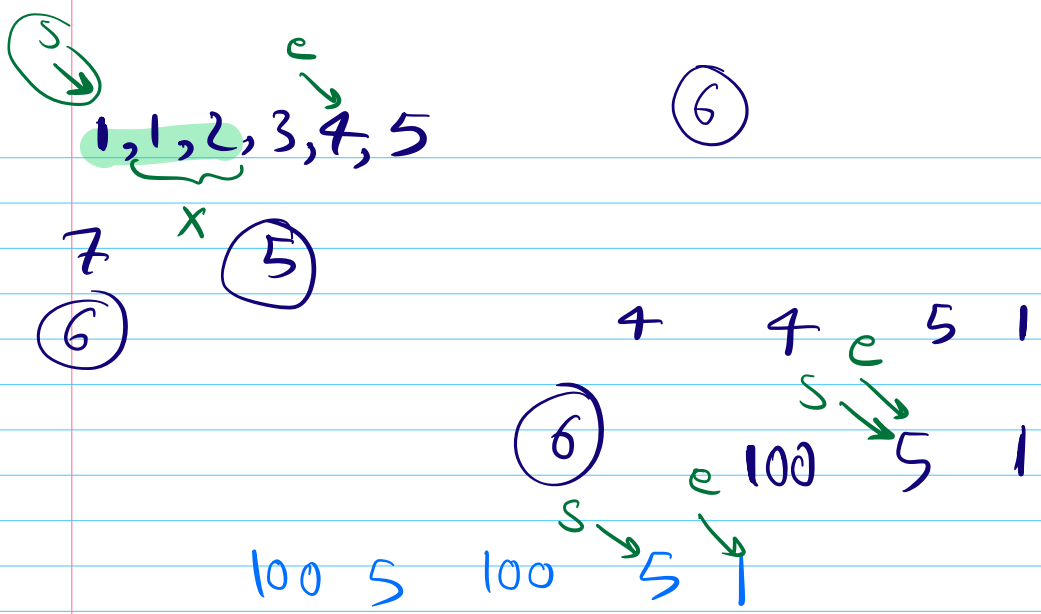
TC: $O(2^n)$

Tree for Fib SC



SC: max size of stack is n not all the possible boxes → max height

(→ O(n))


s
e
1,1,2,3,4,5
x
7
5
6
6
4
4
5
1
s
e
6
100
5
1
e
s
100
5
100
5
1