



Stack
 push $O(1)$
 pop $O(1)$

Queue
 $O(1)$ enqueue — push
 $O(n)$ dequeue — pop

Dequeue

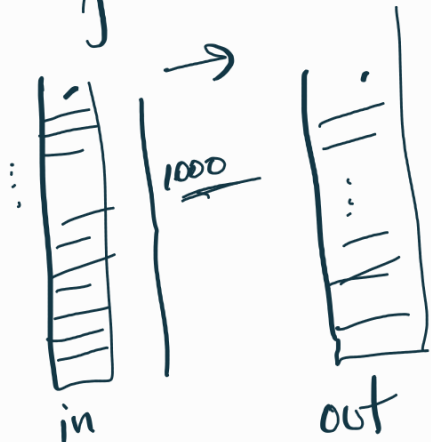
• worst $O(n)$

1,000 elements to put in the queue

- a. 1,000 consecutive enqueues
- b. 1,000 " dequeues

Total operations:

1,000 pushes + 1,000 pops +
 2,000 pushes = 4,000 operations $O(1)$



• best $O(n)$

1,000 elements

1 enqueue → 1 dequeue → 1 enqueue → 1 dequeue
 → ...
 1,000 times.

Queue —

Queue
 stack

best $\xrightarrow{\text{case}} ?$
 worst $\xrightarrow{\text{case}} ?$

$\overline{O(1)} \xrightarrow{n} O(n)$

def Fib(n):

if (n==0) or (n==1):
return n

else:

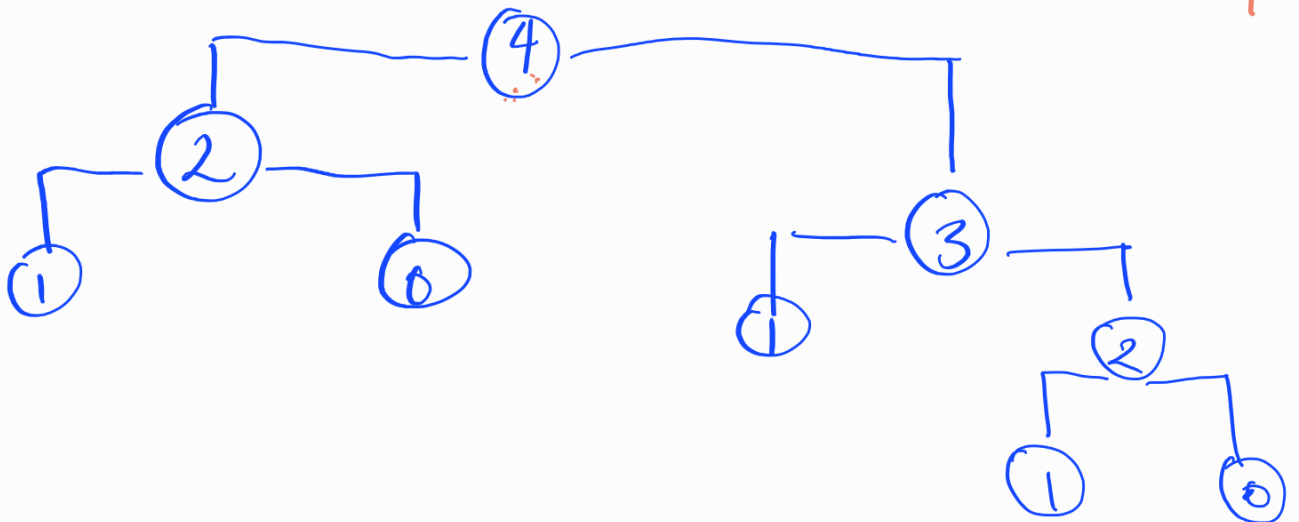
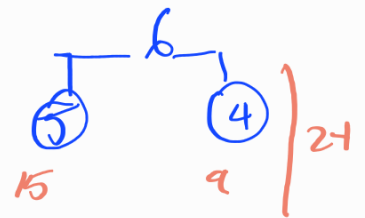
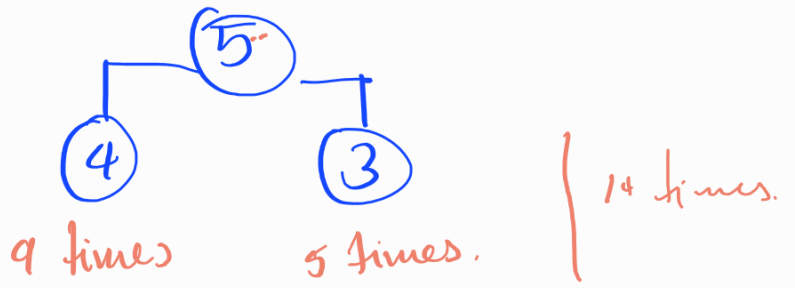
return Fib(n-1) + Fib(n-2)

R: $2^?$ $n-1^?$ $(n-1)(n-2)^?$ $(n-1) + (n-2)^?$
 $O(s) = O(1)$

\overline{no} nested loop exponential

n	R
2	2
3	4
4	8
5	14
6	24

$\sim O(2^n)$

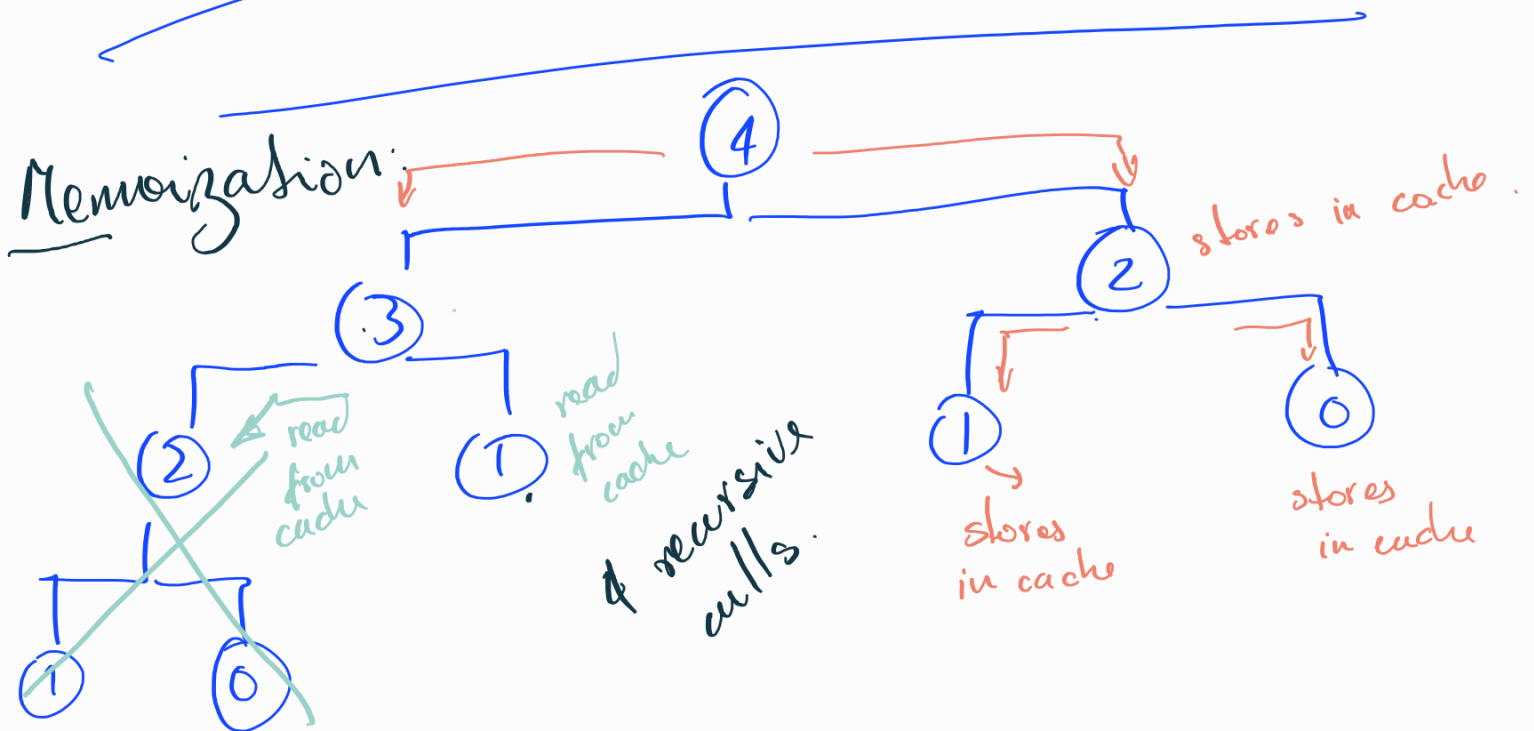


$R = O(2^n)$

$$O(s) = O(1)$$

$$T(n) = O(2^n)$$

Memorization:



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

$$R = O(n)$$

$$O(s) = O(1)$$

$$T(n) = O(n)$$

