Quick Sort ~~qs([7, 2, 5, 4, 8, 9, 6])~~

pivot. [2, 4, 5, 6, 7, 8, 9]

~~qs([2, 4, 5, 6])~~ [7] [8, 9]

~~qs([2, 5, 4, 6])~~ ~~qs([8, 9])~~

↓ qs([]) + [8] + qs([9])

qs([]) + [2] + ~~qs([5, 4, 6])~~ [4, 5, 6]

↓

qs([4]) [5] qs([6])

```
def quicksort(L):
    if len(L) <= 1:
        return L[:]
    else:
        pivot = L[0]
        return quicksort([x for x in L[1:] if x < pivot]) +
               [pivot] +
               quicksort([x for x in L[1:] if x >= pivot])
```

Worst Case.

[1, 2, 3, 4, 5, 6, 7 ... n]

[] qs([2, 3, 4, 5, 6, 7]) } O(n) 层. · O(n) 层.

↓

⋮

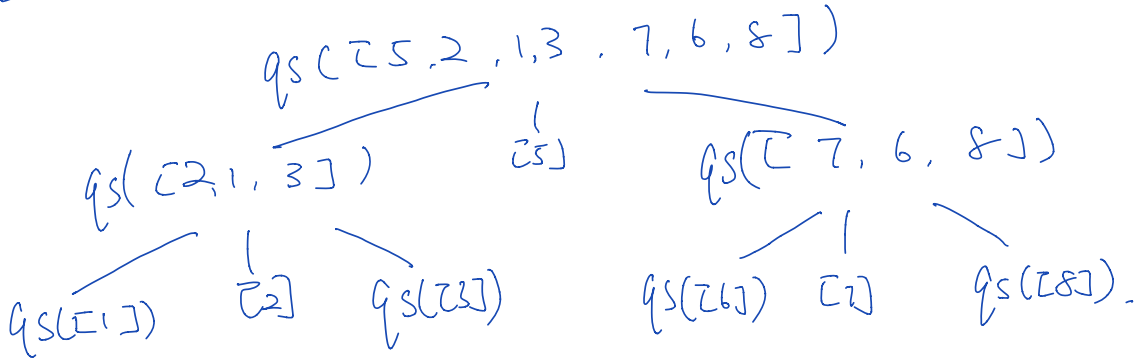
O(n) = O(n²)

[1 7 2 6 3 5 4]

Every pivot picked is either the smallest or the largest of the list.

Best Case:

[5 2 1 3 7 6 8]



$O(\log n)$ 层 $\times$ $O(n)$ / 层 $= O(n \log n)$

Every pivot picked is the median of input list.

[1,2,3,4,5,6]

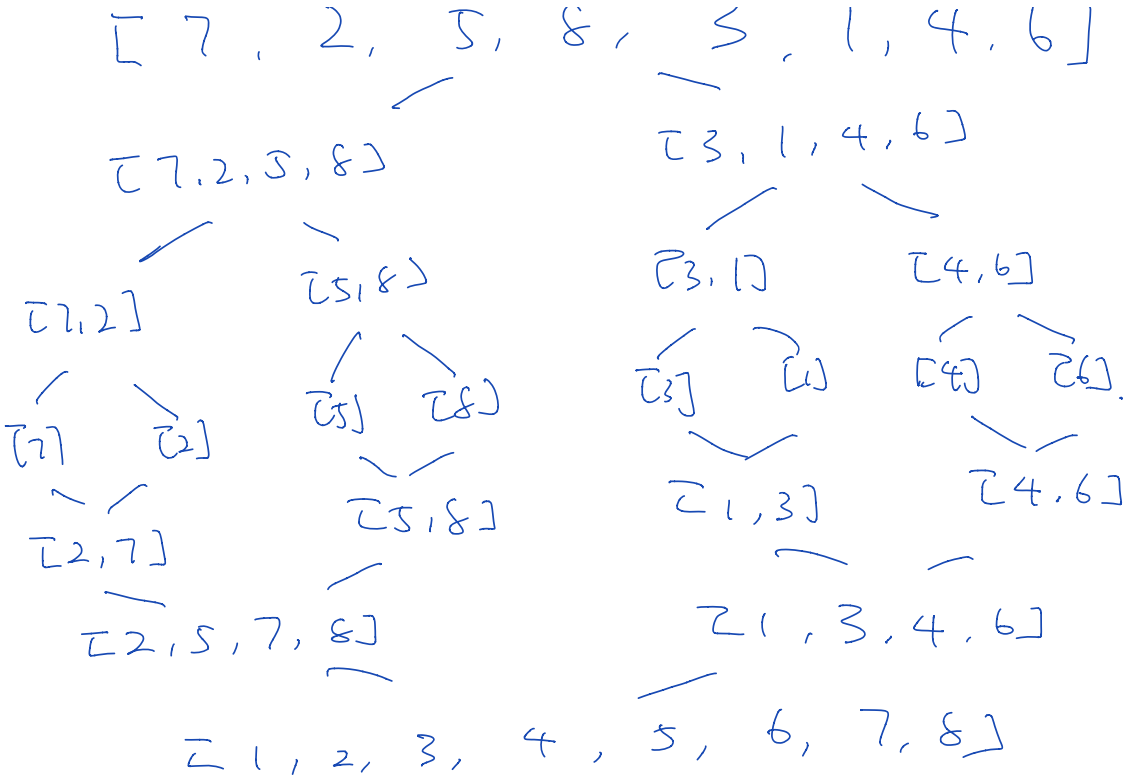
[3,1,2,4,5,6] ✓

Merge Sort: merge. [1,3,7] ⟶ [1,2,3,4,6,7]

[2,4,6]

list 1 size m, list 2 size n.

$O(m+n)$



split: $O(\log n) \cdot O(n)$

$O(n \log n)$

L[a:b] $O(b-a)$

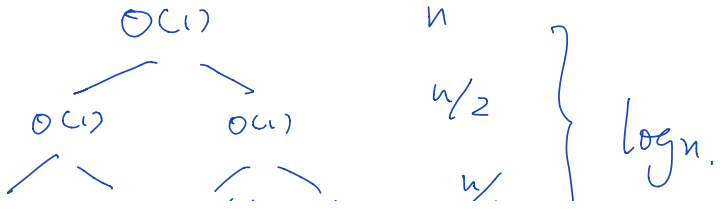
combine: $O(\log n) \cdot O(n)$

$O(n \log n)$

$O(n \log n)$ in total.

assume $n = 2^a$ for $a \in \mathbb{N}^+$

```
def foo(n):
    if n == 1:
        return
    print('haha')
    foo(n//2)
    foo(n//2)
```



foo(n//2) O(1) O(1) O(1) O(1) 4
O(1) . . . . . O(1) . . . O(1) 1

how many O(1)? $2^{\log n} - 1$

$O(n)$

```
def bar(L):
    if len(L) <= 1:
        return
    bar(L[: len(L)//2])
    bar(L[len(L)//2 :])
```

$O(n \log n)$.

memorization

```
def fib(n):
    if n <= 2:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

```
mem = {}    # Global.
def fib(n):
    if n <= 2:
        return 1
```

if n in mem:
    return mem[n]

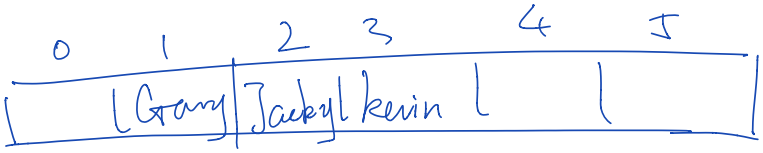
else: result = fib(n-1) + fib(n-2)
mem[n] = result
return result.

Hash table.

key → value.

key ⇝ → constant time get value.

key → hash function → address.

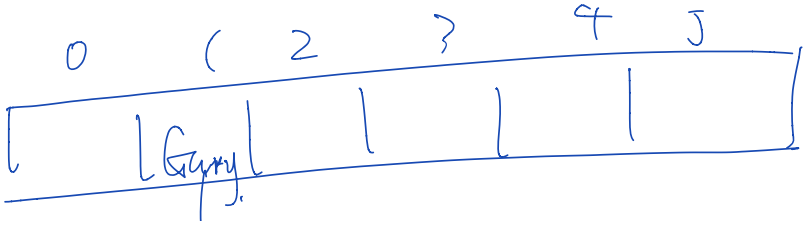


probing

Gary. 1

Jacky. 1.

kevin 2



chaining. Jacky

Gary 1

Jacky 1

① Spel 3+ major. 1

~~321~~ 1 421

458 250 Grael.

3.7 +1

② 1. scrapy

① crawler.
② web program.
③ network.

2. raspberry PI python.

3. the new boston.

4. leetcode.

5. github.com. try.github.io.

blog. csdn