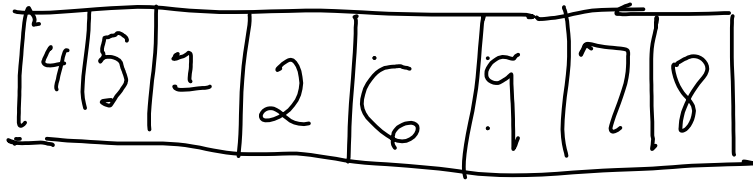


Alg	Time	Space
Insert		$O(1)$
Bub	$O(n^2)$	
Sel		
Merge	$O(n \log n)$	$O(n)$
Quicksort		$O(\log n)$ in-place

---

R L  
↓ ↓

pivot



4 2 1 5 | 6 | 7 9 8

Function PARTITION(A, start, end) :

pivot ← A[end]

pick random index [start, end]

Swap A[index], A[end]

left ← start

right ← end - 1

While left < right:

While A[left] ≤ pivot:

left ← left + 1

While A[right] ≥ pivot:

right ← right - 1  
If left < right:

Swap A[left], A[right]

Swap A[left], A[end]

Return left

Fn Quicksort (A, start, end):

If start  $\geq$  end:

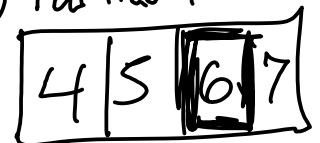
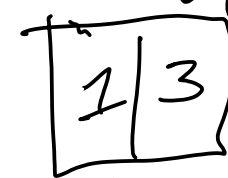
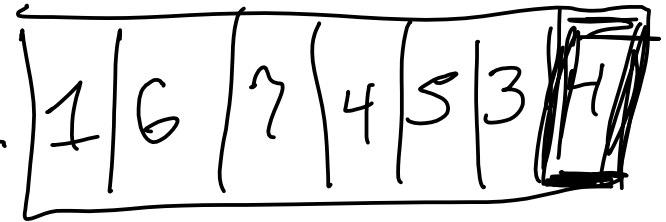
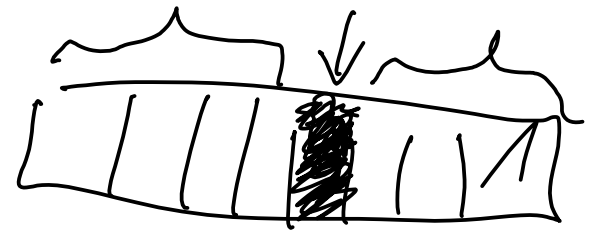
Return

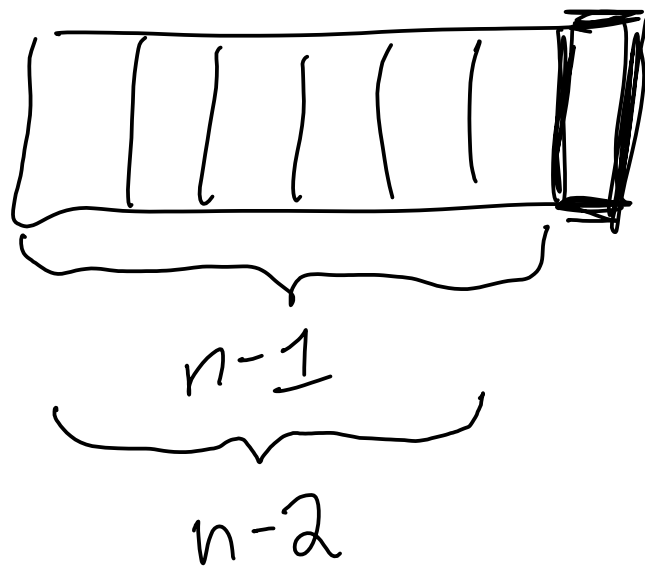
index  $\leftarrow$  Partition(A, start, end)

Quicksort (A, start, index-1)

Quicksort (A, index+1, end)

End Function





Quicksort is  $O(n^2)$

$$\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \dots$$

$$\sum_{n=1}^{\infty} \frac{n}{2^n} = 2$$

CrumbSort (A, n):  
Flip a coin  
until heads

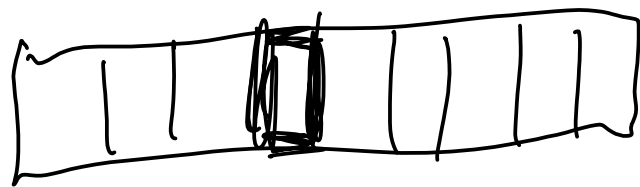
MagicSort(A, n)

	Luck	Adversary
worst	Bad	Bad
worst expect	Normal	Bad

“worst” case  
“best” case  
expected case

→ expected worst case

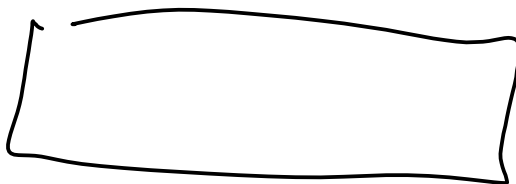
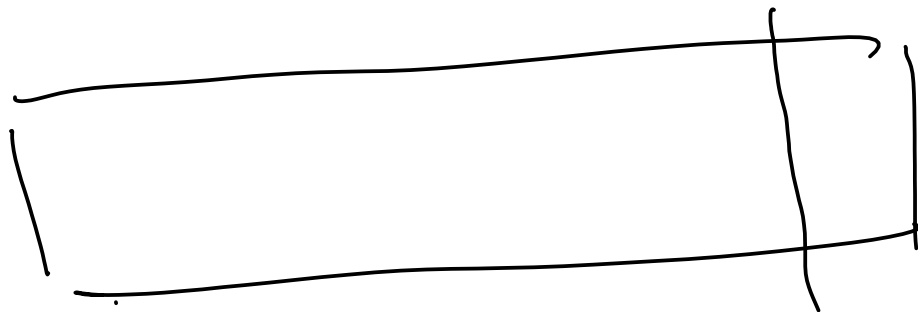
Random  
pivot



FWC

$$O(n \log n)$$

99th



99%



1%



⋮

$$\left. \begin{array}{l} \text{Diagram 1} \\ \text{Diagram 2} \\ \text{Diagram 3} \\ \text{Diagram 4} \\ \text{Diagram 5} \end{array} \right\} \log_{\frac{100}{99}} n$$