SORTING

Bubble Sort

[4 8 3 1 2]

4 3 8 1 2

4 3 1 8 2

4 3 1 2 8

---

3 1 2 4 8

---

1 2 3 4 8

[4 | 8 3 1 2]

4 8 3 1 2

Insertion

traverse through sorted to find spot

Selection Sort

[4 3 8 1 2]

[1 3 8 4 2]

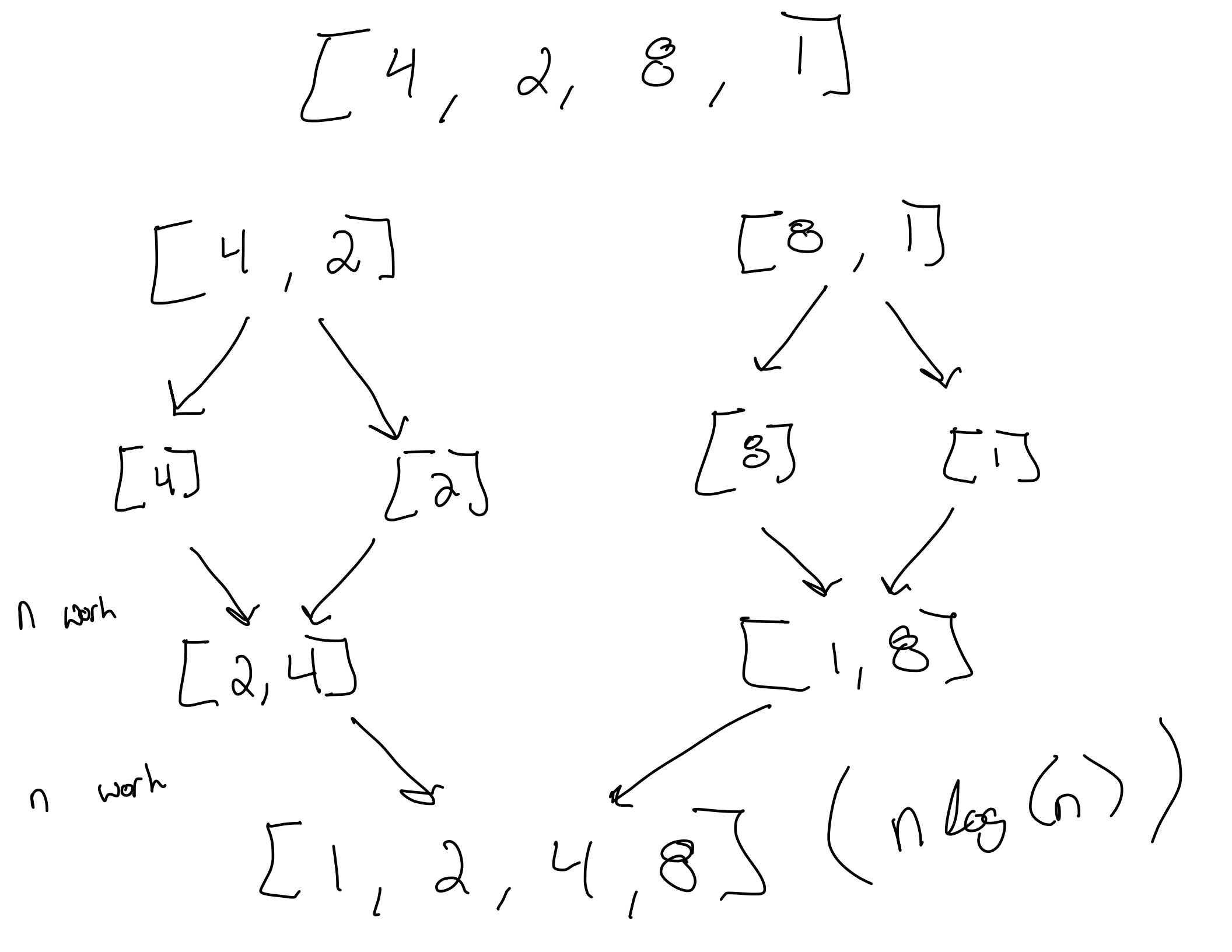
[1 2 8 4 3]

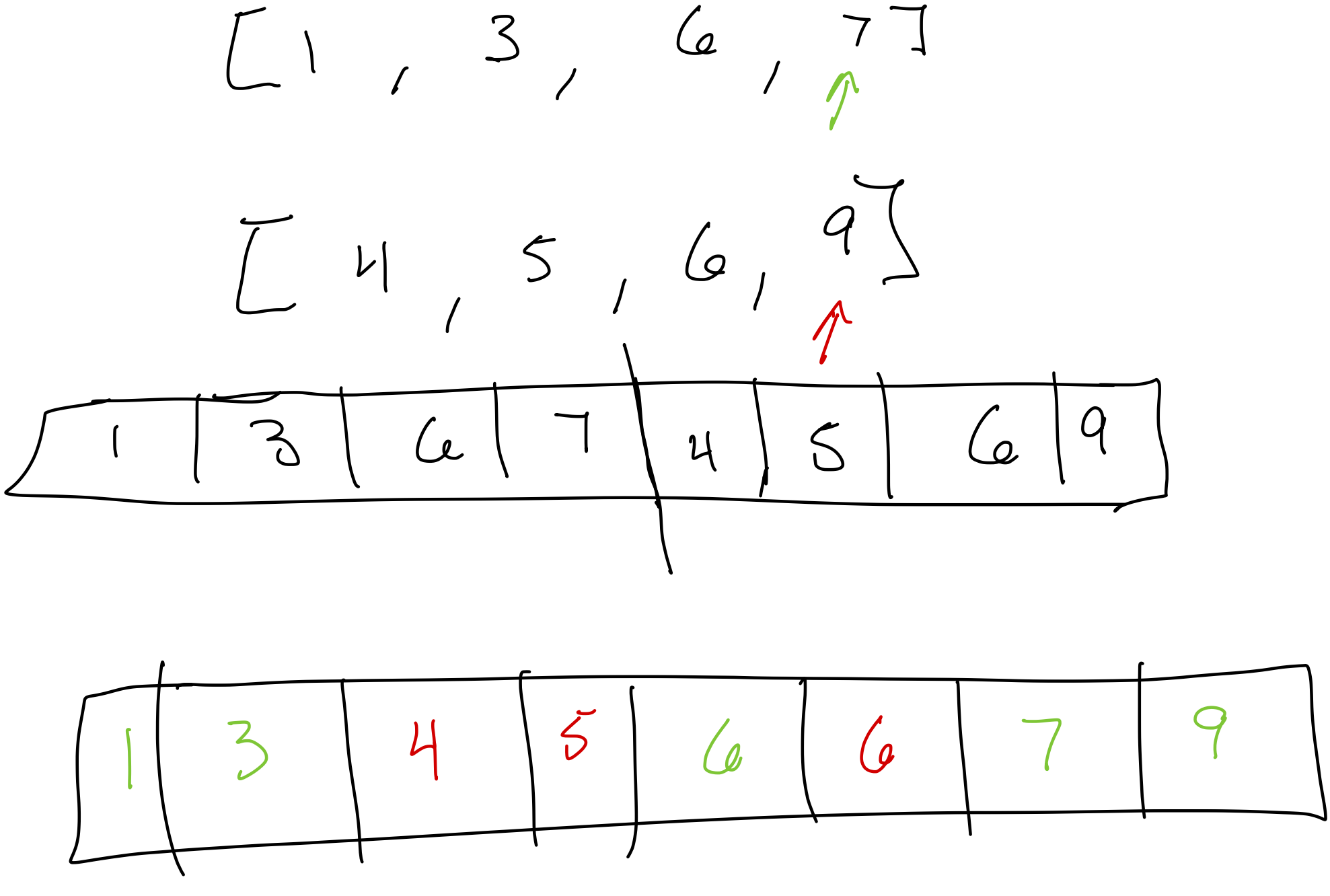
[1 2 3 4 8]

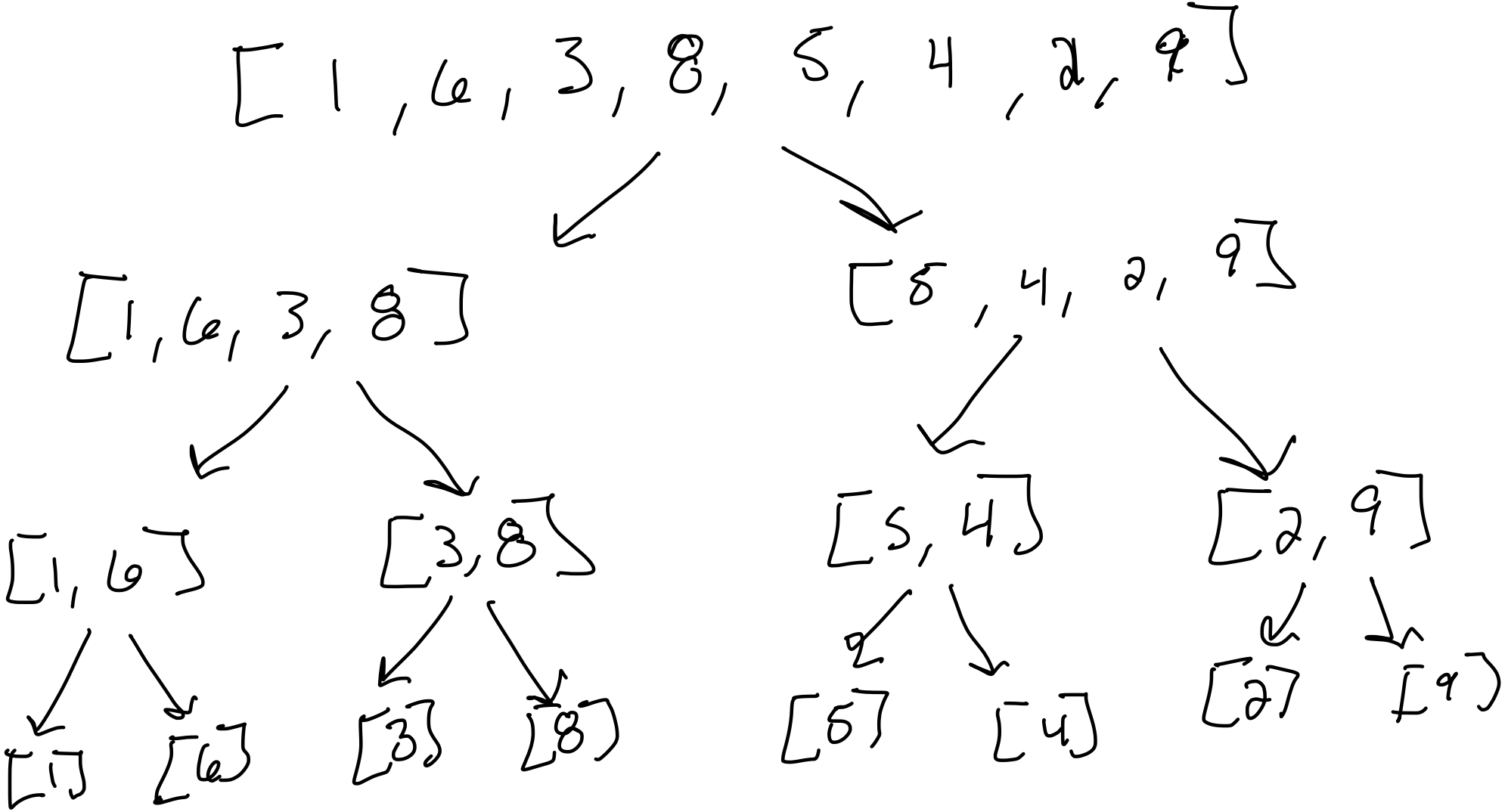
traverse through unsorted to find next smallest

$O(1)$ $O(n)$ $O(n^2)$

$O(n^3)$


[4, 2, 8, 1]
[4, 2]
[8, 1]
[4]
[2]
[8]
[1]
n work
[2, 4]
[1, 8]
n work
[1, 2, 4, 8]
(n log (n))


[1, 3, 6, 7]
[4, 5, 6, 9]
1 3 6 7 4 5 6 9
1 3 4 5 6 6 7 9


[1, 6, 3, 8, 5, 4, 2, 9]
[1, 6, 3, 8]
[5, 4, 2, 9]
[1, 6]
[3, 8]
[5, 4]
[2, 9]
[1]
[6]
[3]
[8]
[5]
[4]
[2]
[9]

MERGE SORT

- divide and conquer algorithm
- recursive
- process:
  - divide array into 2 halves
  - **recursively** sort each half (by calling mergesort on each half)
  - merge sorted halves (take 2 sorted lists and combine into one sorted list)

MERGE SORT - COMPLEXITY

$\log_2(n)$

- Tree like, recursive halving/combining
- How much work at each step of tree?

n

time complexity: $O(n \log(n))$

Space complexity: $O(n)$

QUICK SORT

- divide and conquer algorithm
- recursive
- process:
  - choose pivot (often leftmost or rightmost element)
  - place:
    - everything less than pivot to it's left
    - everything greater than pivot to it's right
    - pivot in between
  - repeat process for left chunk and right chunk (use quicksort to sort each chunk)

QUICK SORT

- How to move elements to correct side of pivot?
- Start scanning from both sides
  - Scan from left -> until find entry > pivot
  - Scan from <- right until find entry < pivot
  - Swap entries
- Stop once scan indices cross and swap pivot element into place
  - leftmost pivot, swap pivot with rightmost in left subarray
  - rightmost pivot, swap pivot with leftmost in right subarray

[34 30 62 28 27 9 84 33 67]

[9 30 33 28 27 34 84 62 67

QUICK SORT - COMPLEXITY

- Recursive halving of problems - how many levels?
- How much work at each level

	best	average	worst	space complexity	stable
quicksort	$n\log(n)$	$n\log(n)$	$O(n^2)$	$\log(n)$	no
mergesort	$n\log(n)$	$n\log(n)$	$n\log(n)$	n	yes
bubble sort	n	n^2	n^2	1	yes
insertion sort	n	n^2	n^2	1	yes
selection sort	n^2	n^2	n^2	1	no