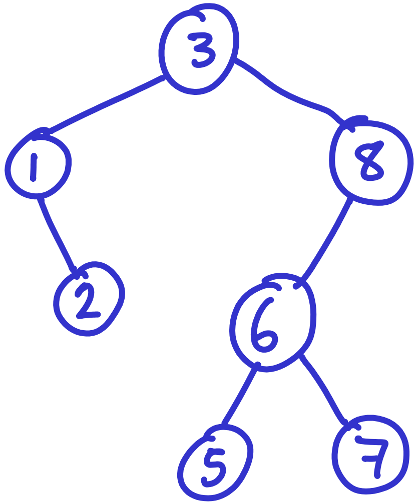
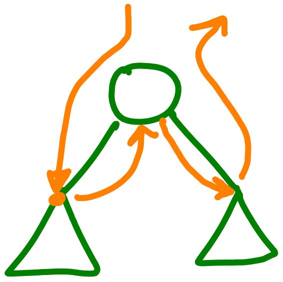
BINARY SEARCH TREES – BUILT RANDOMLY

We want to insert n elements into a BST
so that they will be stored in sorted order

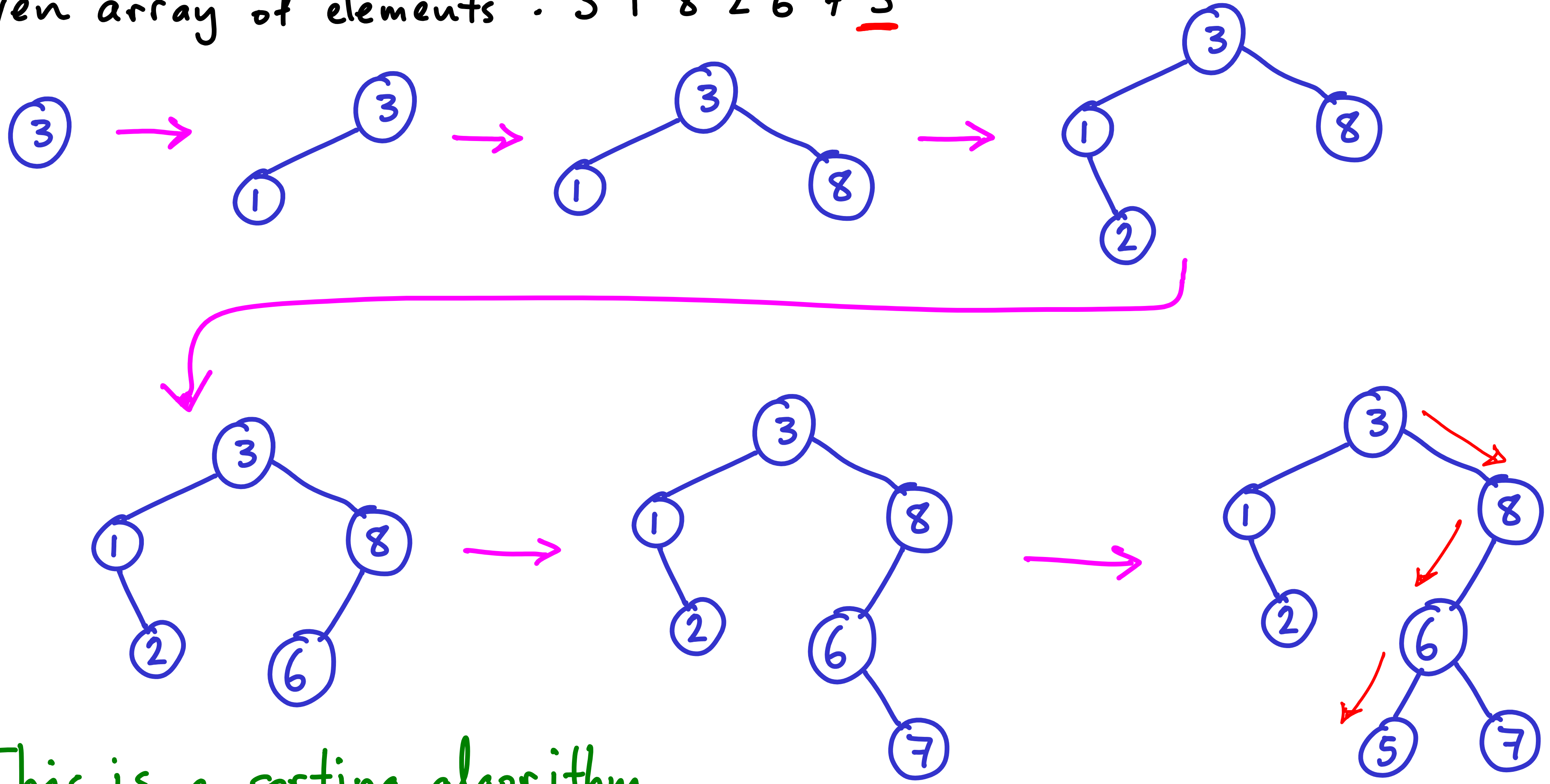


InOrder walk : 1 2 3 5 6 7 8



Insertion : nothing fancy.
Just read elements and insert
into current tree.

Given array of elements : 3 1 8 2 6 7 5 write down



This is a sorting algorithm.

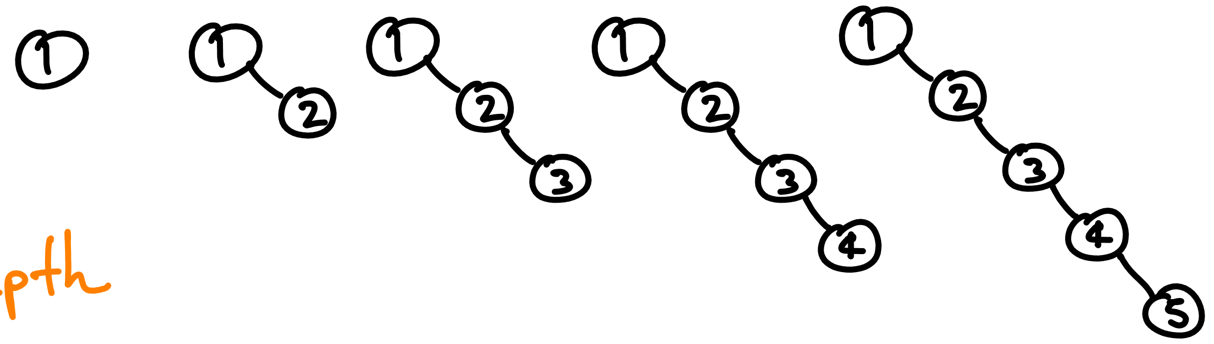
write down

Given the very simple BST-sort/construction algorithm

- how long can it take to build the BST?
- what shape will it have? How balanced? What depth?

Depends on input sequence

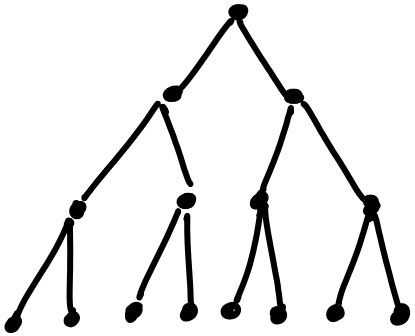
1 2 3 4 5 $\longrightarrow$



Can be bad: $O(n)$ depth

$O(n^2)$ time

$\Omega(n \log n)$ worst-case time: sorting lower bound



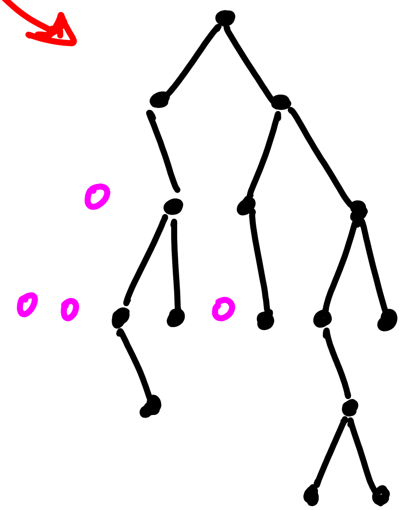
Even for a balanced tree,

$\Theta(n) \approx \frac{n}{2}$ nodes have height $= \Theta(\log n)$

so it must take $\Omega(n \log n)$ time to build.

Any algorithm producing any tree shape : $\Omega(n \log n)$ time

worse



So ... if lucky, $\Theta(n \log n)$ time

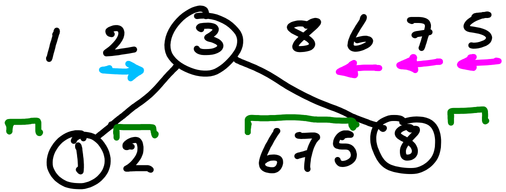
if unlucky, $O(n^2)$ time

Sounds like ... quicksort

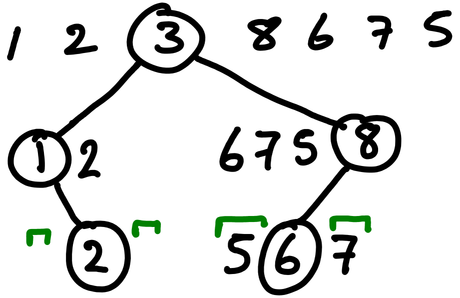
Stable quicksort

3 1 8 2 6 7 5

- use first elt to partition → 1 2 3 8 6 7 5
- repeat on each side

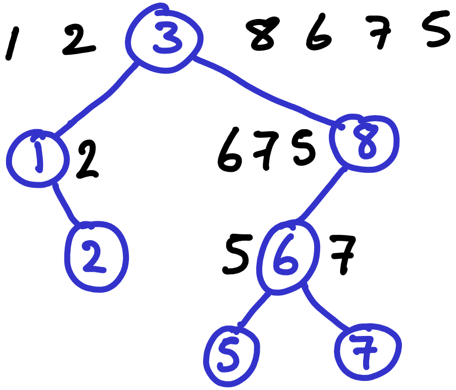


- 3rd round



- 4th round

same tree as BST



quicksort round 1: compare all elts to 3

BST sort : 3 = root; eventually all elts pass through.

quicksort : partitions into 2 groups
<3 & >3
each is independent

BSTsort : same

exactly same comparisons but in different order

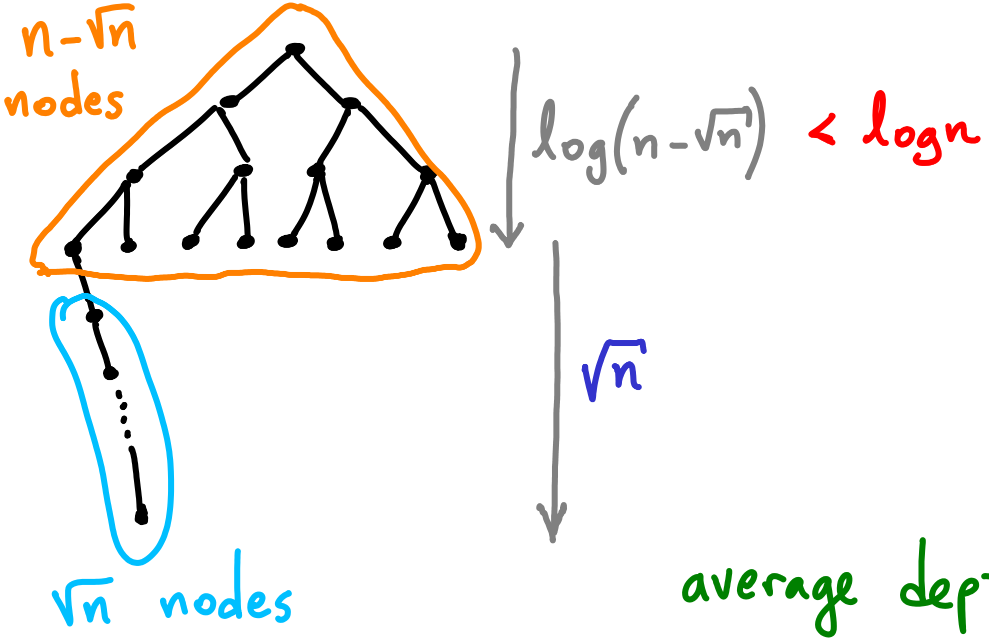
We've seen QUICKSORT $\simeq$ BST-SORT $\Big\}$ Randomized versions have same analysis

(stable)

$E[\text{BST build}] = \Theta(n \log n)$ time

$E[\text{insert node}] = \Theta(\log n)$ time

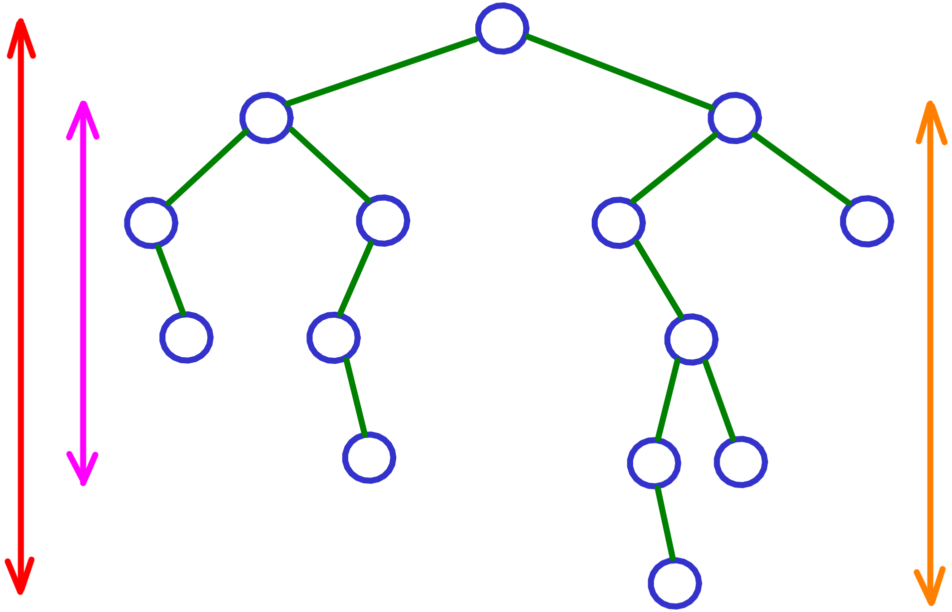
Intuition: $E[\text{depth}] = \Theta(\log n)$ so it should be ~balanced? NO



$$\text{average depth} < \frac{1}{n}\cdot\left(\overset{\text{exaggerated \#nodes}}{n}\cdot\overset{\text{exaggerated depth}}{\log n} + \overset{\text{\#nodes}}{\sqrt{n}}\cdot\overbrace{(\sqrt{n}+\log n)}\right)$$

$$= \log n + 1 + \frac{\log n}{\sqrt{n}}$$

$$= O(\log n)$$ so $E[\text{depth}] \not\Rightarrow$ balance



$$H(n) = 1 + \max\{H(k), H(n-k-1)\}$$

for some $\underbrace{\text{random } k}_{\text{with rigorous analysis can get } \sim 3\log n}$ $(0 \le k \le n-1)$

If $\frac{n}{4} < k < \frac{3n}{4}$

$\hookrightarrow H(n) \le 1 + H\left(\frac{3n}{4}\right)$

else $H(n) \le 1 + H(n-1)$

$< 1 + H(n)$

$$E[H(n)] \le 1 + \frac{1}{2}E\left[H\left(\frac{3n}{4}\right)\right] + \frac{1}{2}E[H(n)]$$

$$\frac{1}{2}E[H(n)] \le 1 + \frac{1}{2}E\left[H\left(\frac{3n}{4}\right)\right]$$

$$E[H(n)] \le 2 + E\left[H\left(\frac{3n}{4}\right)\right] = O(\log n)$$

$2\log_{4/3} n \sim= 2 * 2.4 \log_2 n < 5\log n$