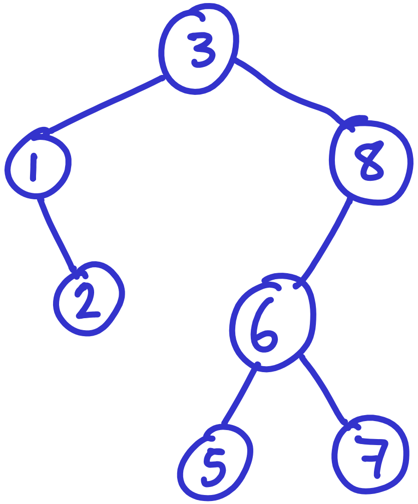
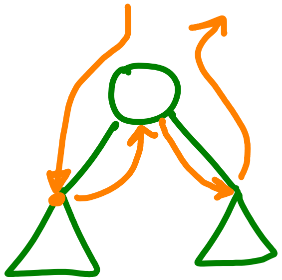
BINARY SEARCH TREES – BUILT RANDOMLY

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

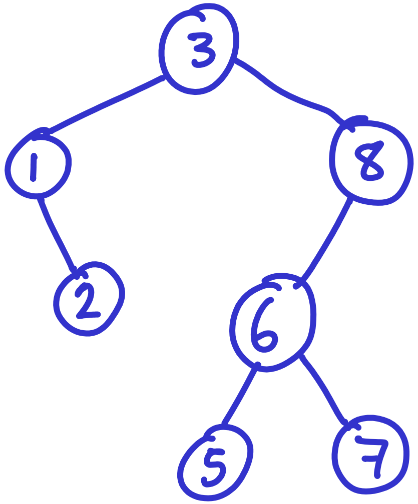


InOrder walk : 1 2 3 5 6 7 8

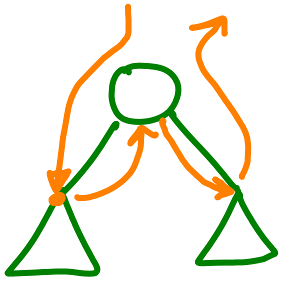


BINARY SEARCH TREES – BUILT RANDOMLY

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

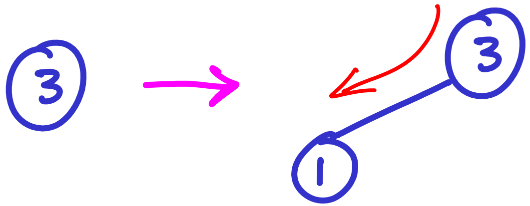


Insertion : nothing fancy.
Just read elements and insert
into current tree.

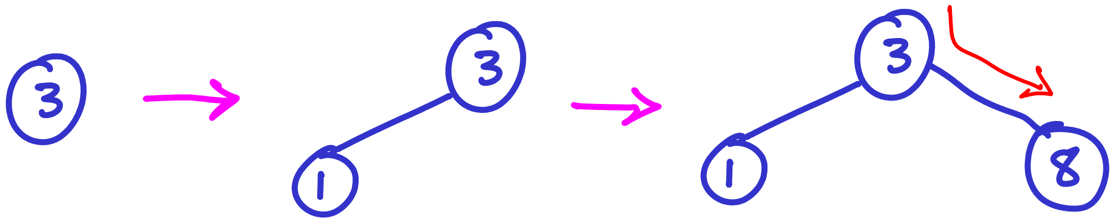
Given array of elements : 3 1 8 2 6 7 5

3

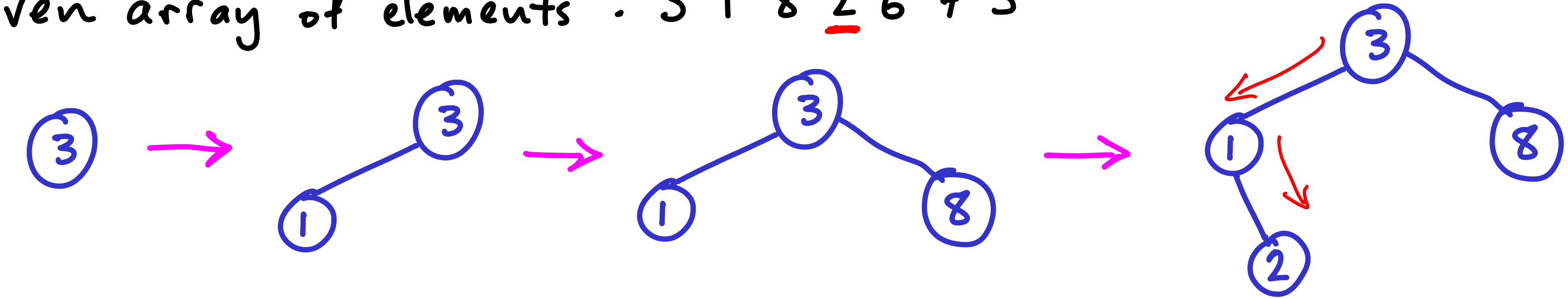
Given array of elements : 3 1 8 2 6 7 5



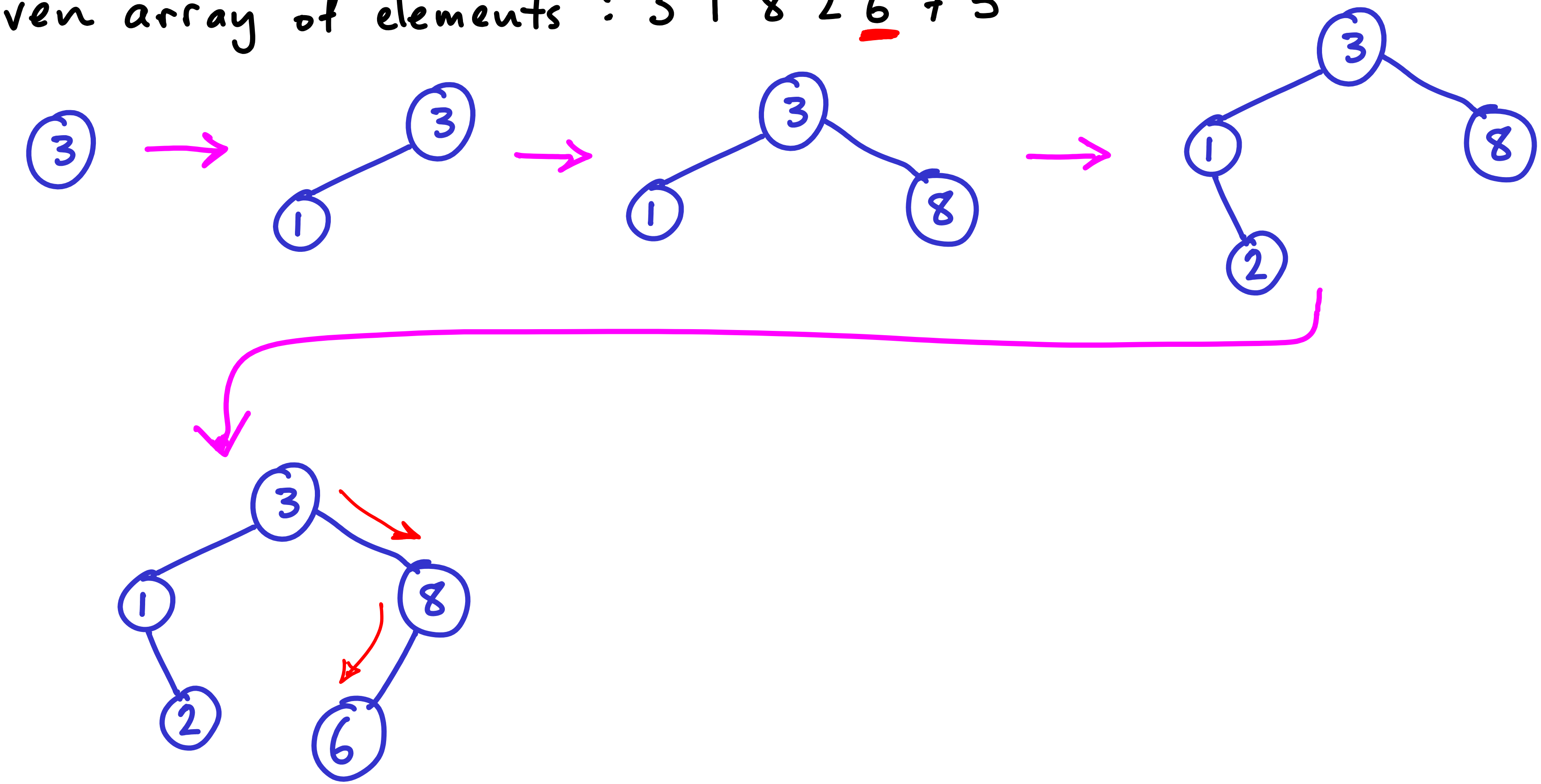
Given array of elements : 3 1 8 2 6 7 5



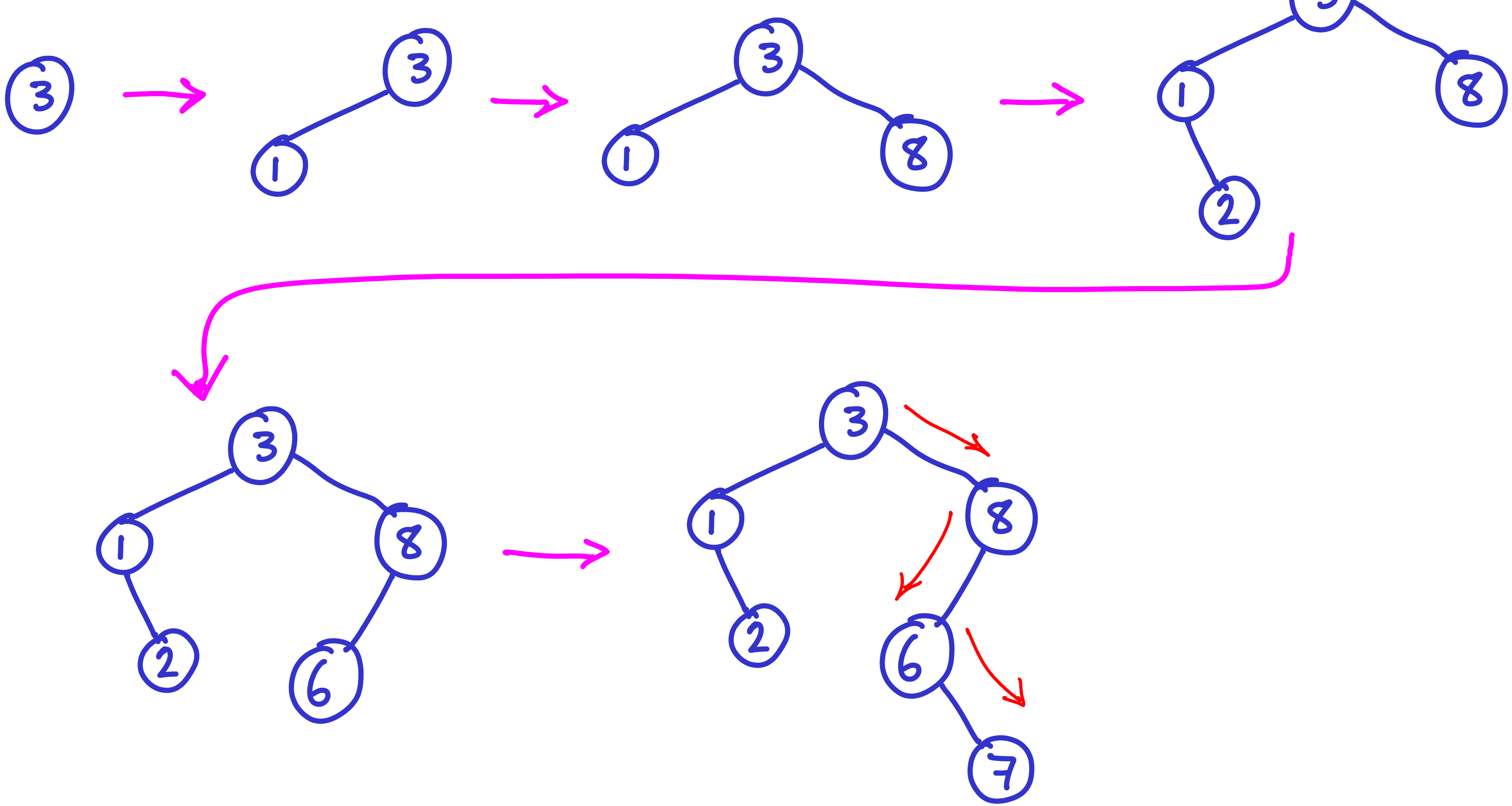
Given array of elements : 3 1 8 2 6 7 5



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

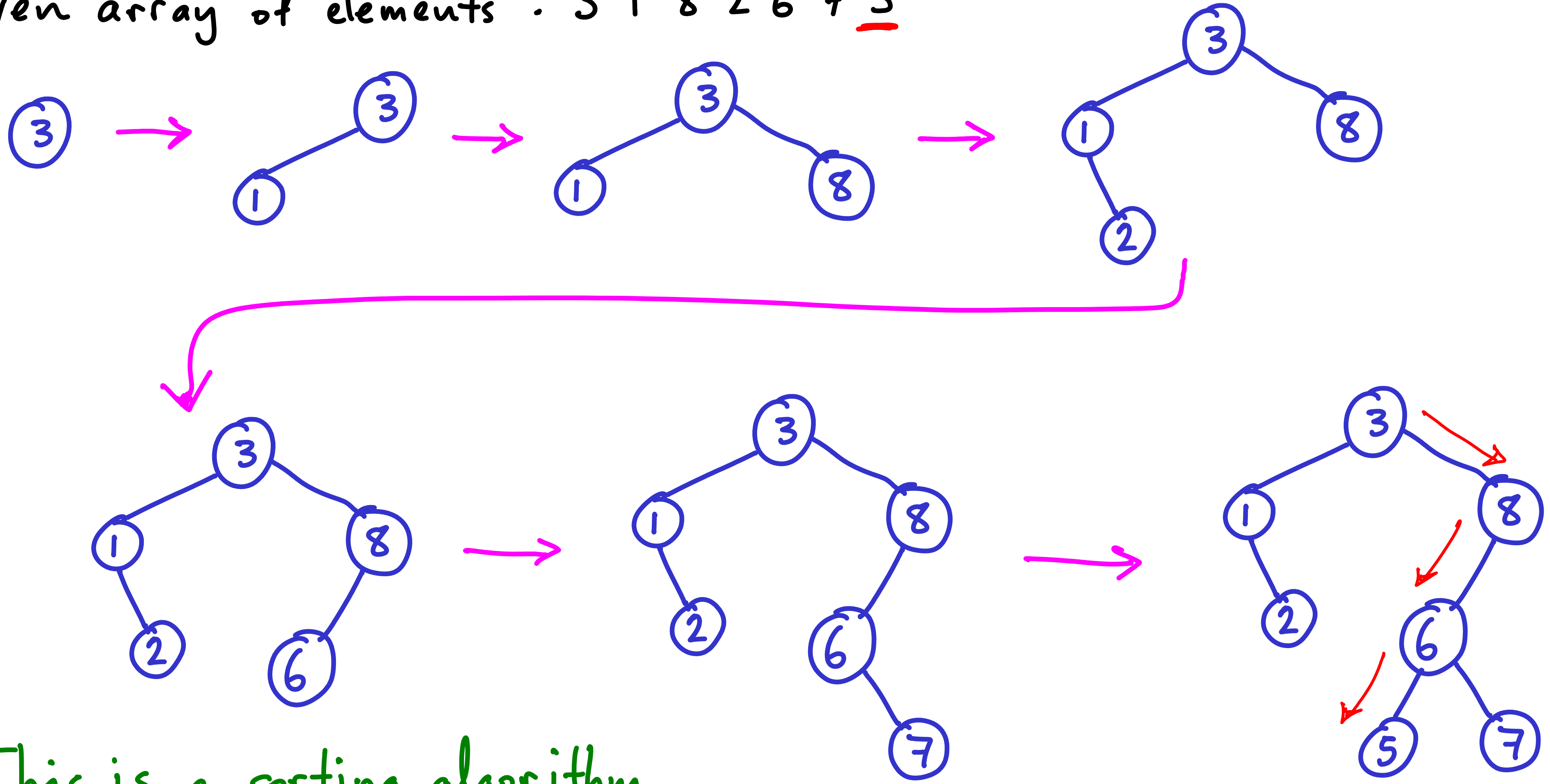


Given array of elements : 3 1 8 2 6 7 5



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

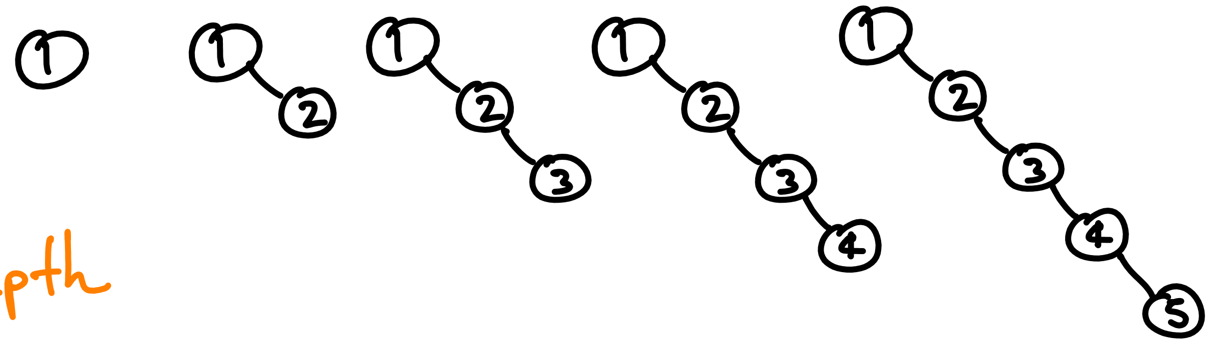
- how long can it take to build the BST?
- what shape will it have? How balanced? What depth?

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

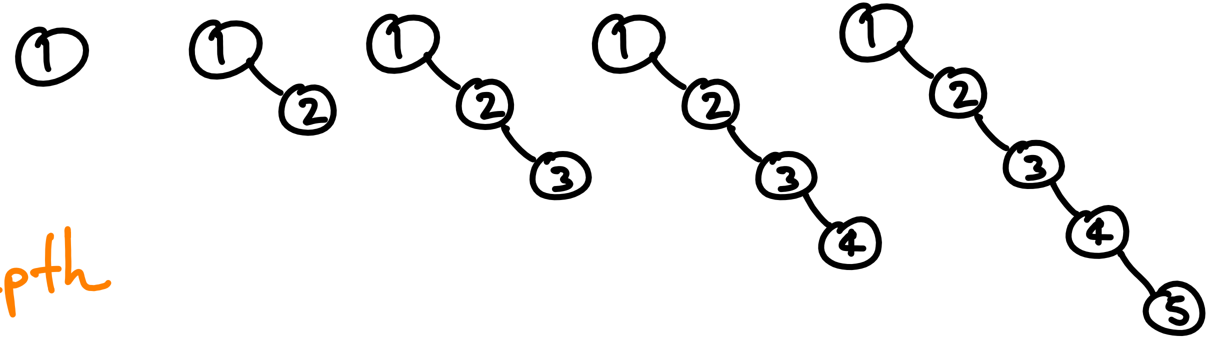
$\Omega(?)$

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

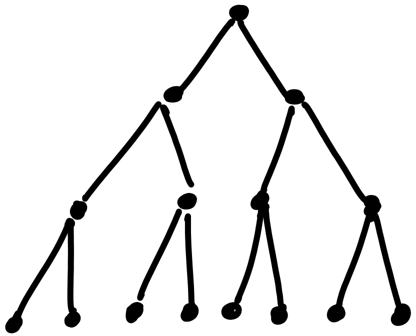
1 2 3 4 5 $\longrightarrow$



Can be bad: $O(n)$ depth

$O(n^2)$ time

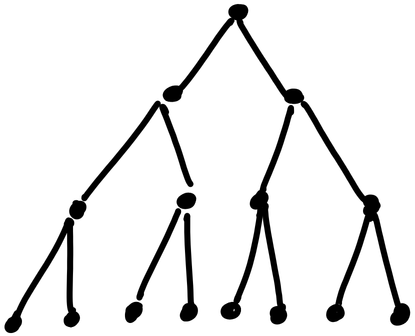
$\Omega(n\log n)$ worst-case time: sorting lower bound



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.



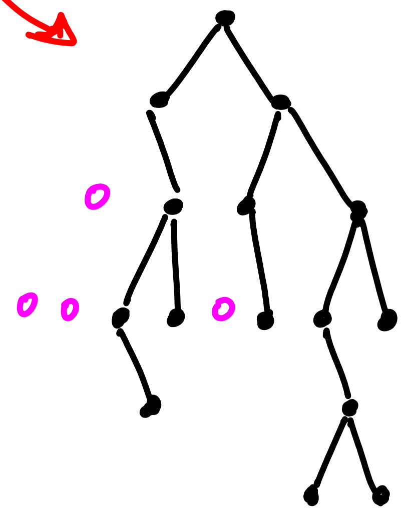
Even for a balanced tree,

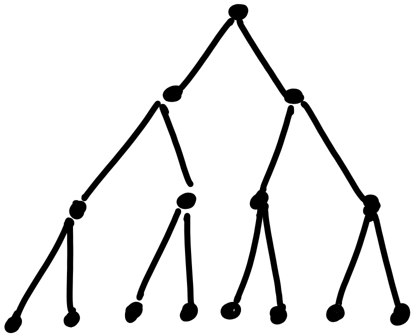
$\Theta(n) \approx \frac{n}{2}$ nodes have height $= \Theta(\log n)$

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

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

worse





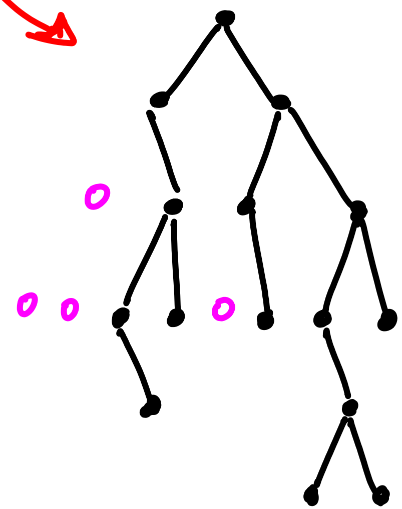
Even for a balanced tree,

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

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

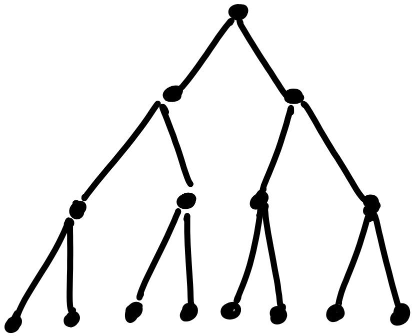
Any algorithm producing any tree shape : $\Omega(n \log n)$ time

worse



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

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



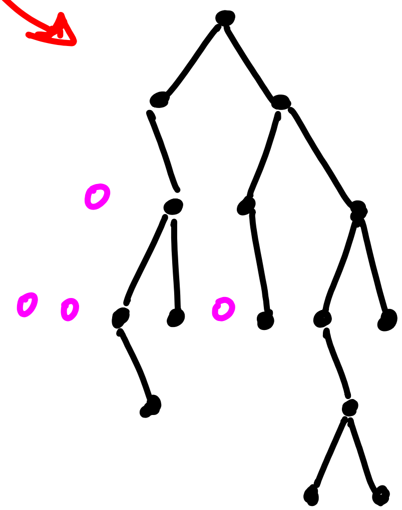
Even for a balanced tree,

$\Theta(n) \approx \frac{n}{2}$ 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

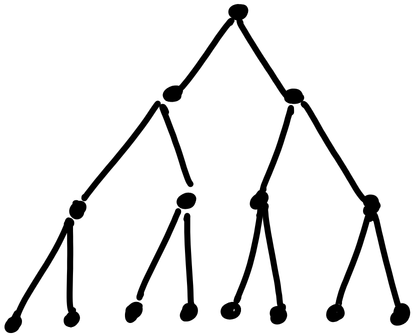
worse



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

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

Sounds like...

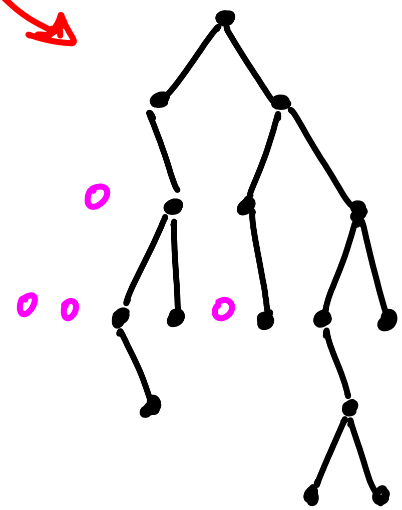


Even for a balanced tree,

$\Theta(n) \approx \frac{n}{2}$ 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



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

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

1 2 3 8 6 7 5

1 2 6 7 5 8

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

1 2 3 8 6 7 5

1 2 6 7 5 8

- 3rd round

1 2 3 8 6 7 5

1 2 6 7 5 8

2 5 6 7

Stable quicksort

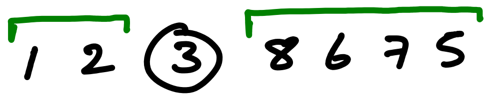
3 1 8 2 6 7 5

- use first elt to partition → 1 2 3 8 6 7 5
- repeat on each side
  - 1 2 3 8 6 7 5
  - 1 2   6 7 5 8
- 3rd round
  - 1 2 3 8 6 7 5
  - 1 2   6 7 5 8
  - 2   5 6 7
- 4th round
  - 1 2 3 8 6 7 5
  - 1 2   6 7 5 8
  - 2   5 6 7
  - 5   7

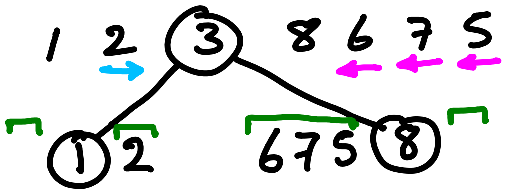
Stable quicksort

3 1 8 2 6 7 5

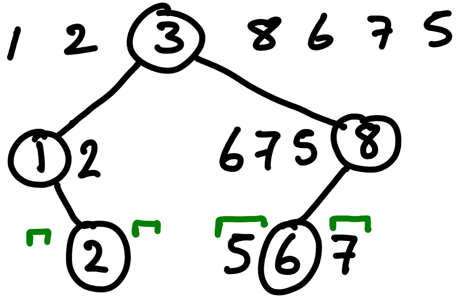
- use first elt to partition →



- repeat on each side

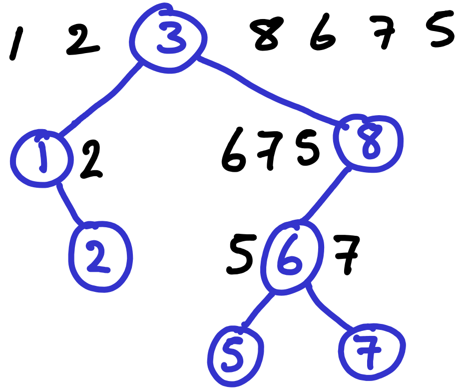


- 3rd round



- 4th round

same tree as BST

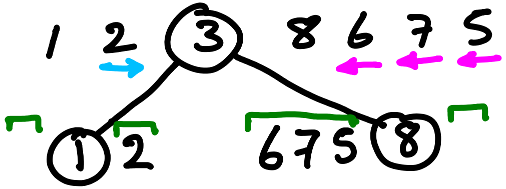


Stable quicksort

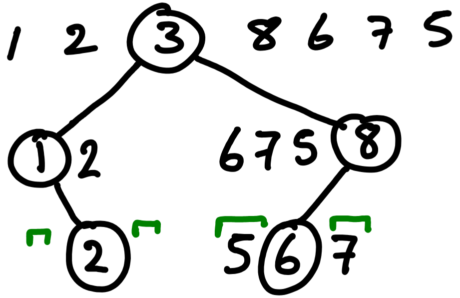
3 1 8 2 6 7 5

- use first elt to partition → 1 2 3 8 6 7 5

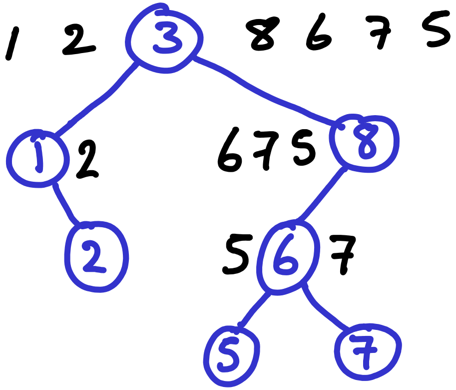
- repeat on each side



- 3rd round



- 4th round



same tree as BST

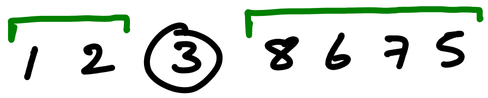
quicksort round 1: compare all elts to 3

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

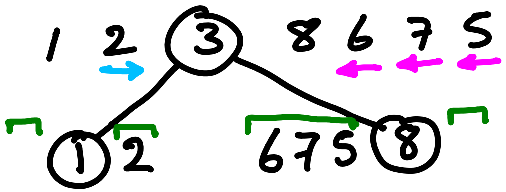
Stable quicksort

3 1 8 2 6 7 5

- use first elt to partition



- repeat on each side



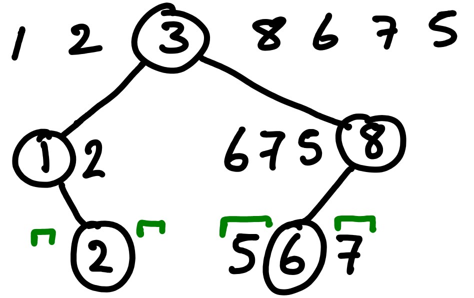
quicksort round 1: compare all elts to 3

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

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

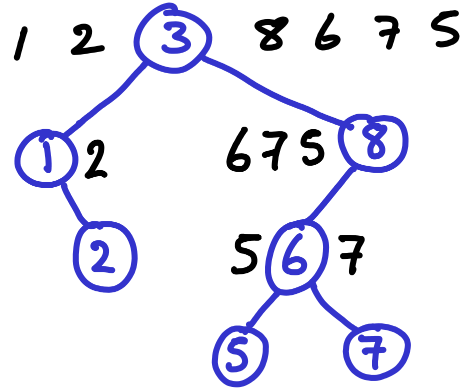
BST sort : same

- 3rd round



- 4th round

same tree as BST

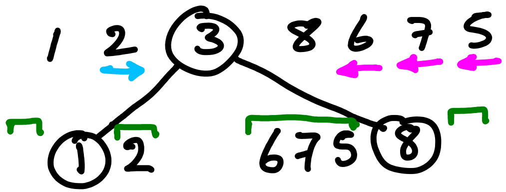


Stable quicksort

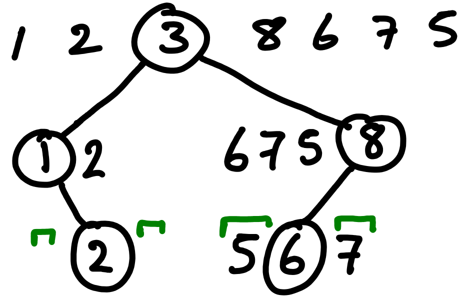
3 1 8 2 6 7 5

- use first elt to partition → 1 2 3 8 6 7 5

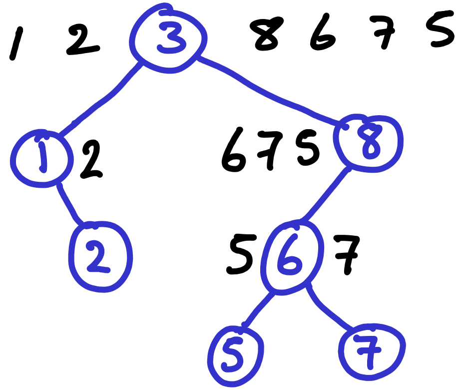
- repeat on each side



- 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

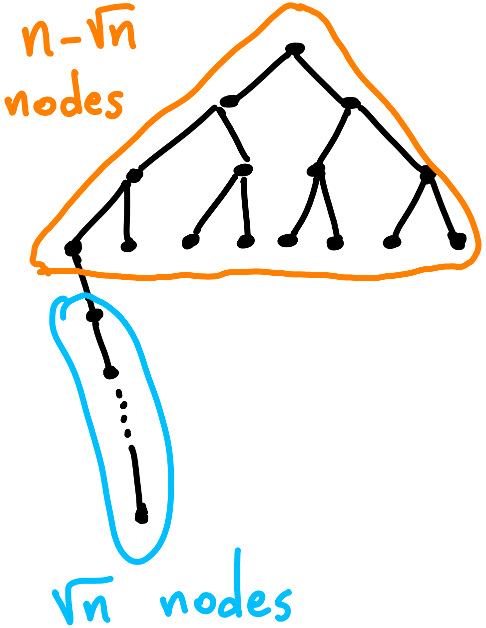
We've seen QUICKSORT $\simeq$ BST-SORT } Randomized versions have same analysis
(stable)

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

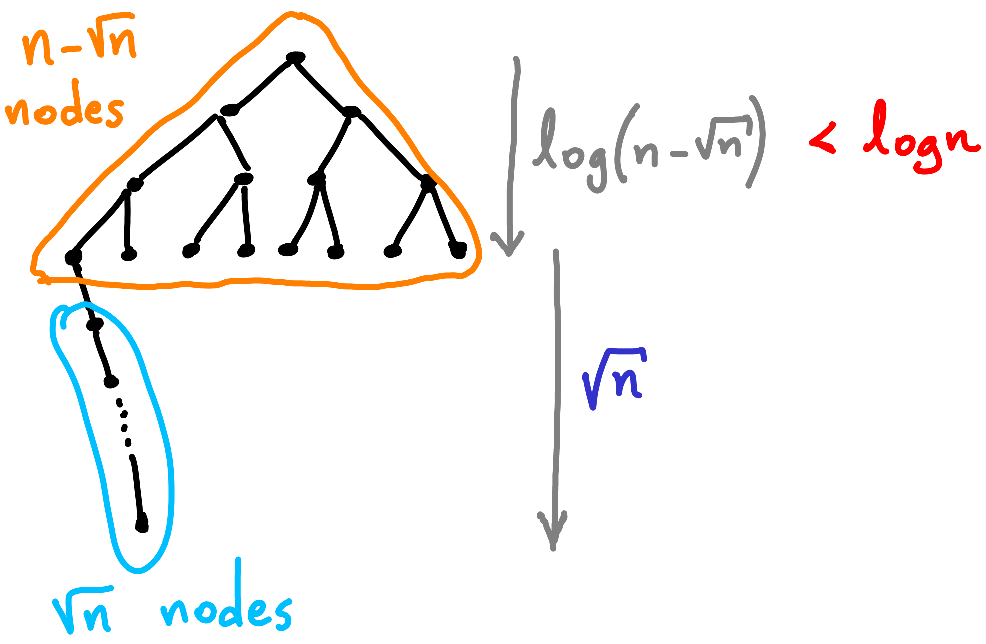
$E[\text{insert node}] = \Theta(\log n)$ time

Intuition : $E[\text{depth}] = \Theta(\log n)$ so it should be ~balanced ?

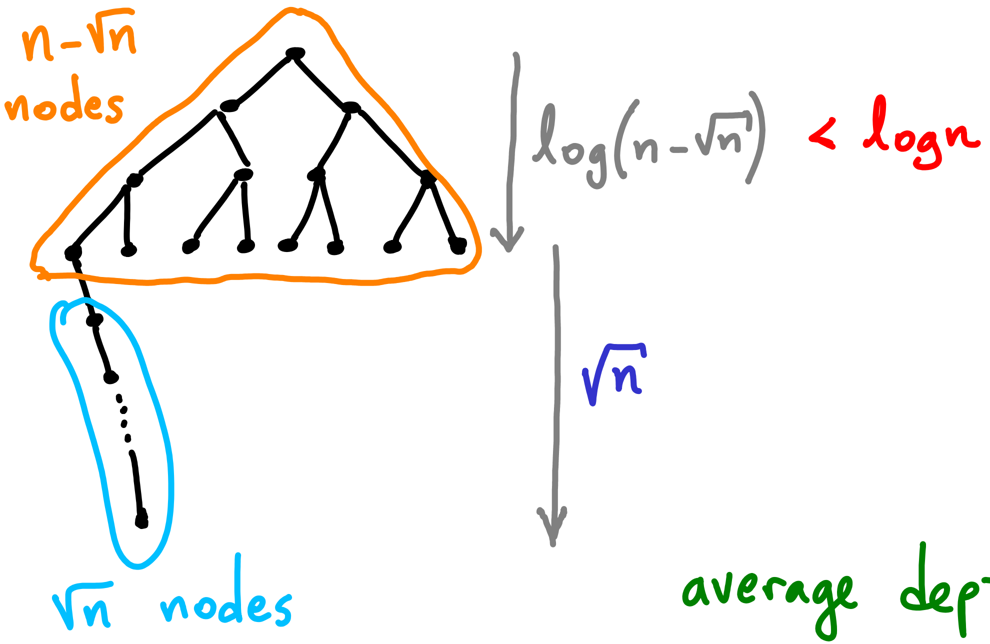
Intuition : $E[\text{depth}] = \Theta(\log n)$ so it should be ~balanced ? NO



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

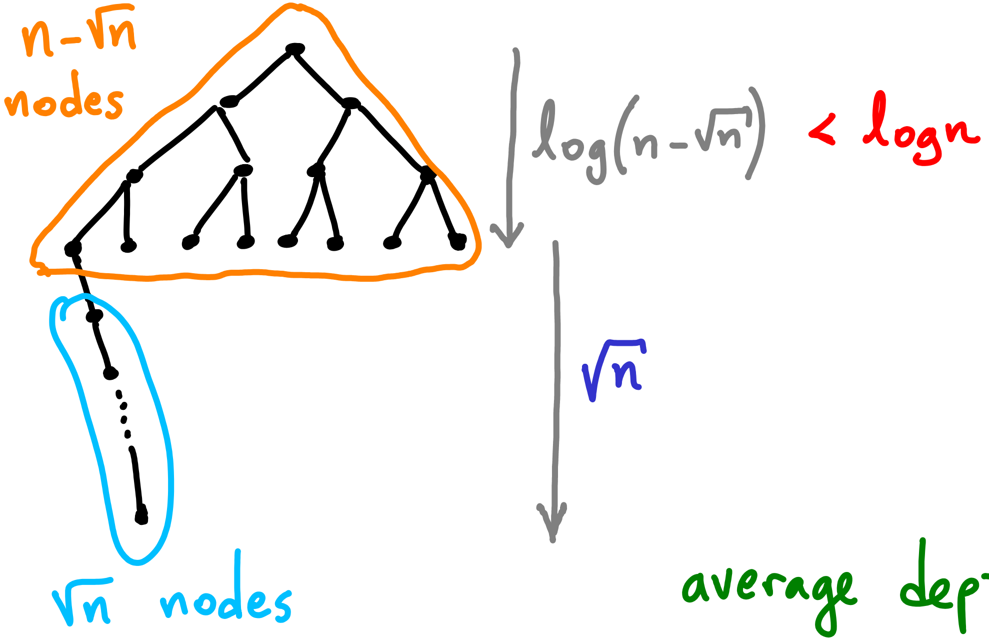


Intuition : $E[\text{depth}] = \Theta(\log n)$ so it should be ~balanced ? NO



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

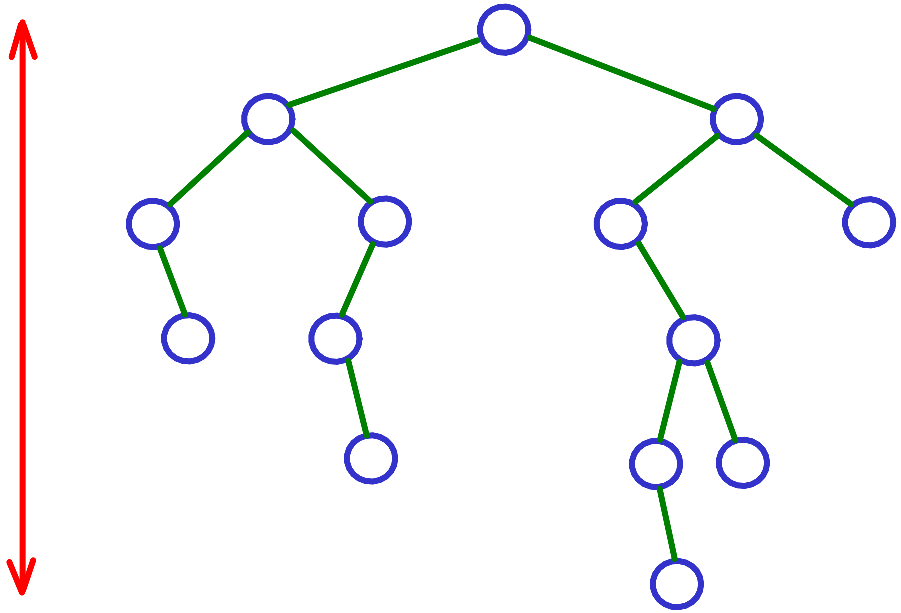
Intuition : $E[\text{depth}] = \Theta(\log n)$ so it should be ~balanced ? NO



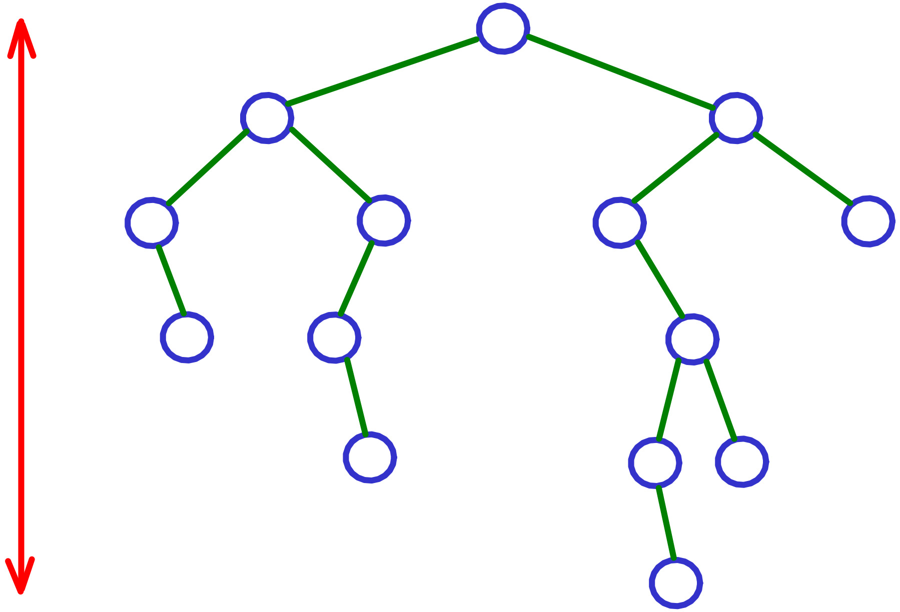
$$\text{average depth} < \frac{1}{n}\cdot\left(\overbrace{n}^{\text{exaggerated \#nodes}}\cdot\overbrace{\log n}^{\text{exaggerated depth}} + \overbrace{\sqrt{n}}^{\#\text{nodes}}\cdot\overbrace{(\sqrt{n}+\log n)}^{\text{exaggerated depth}}\right)$$

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

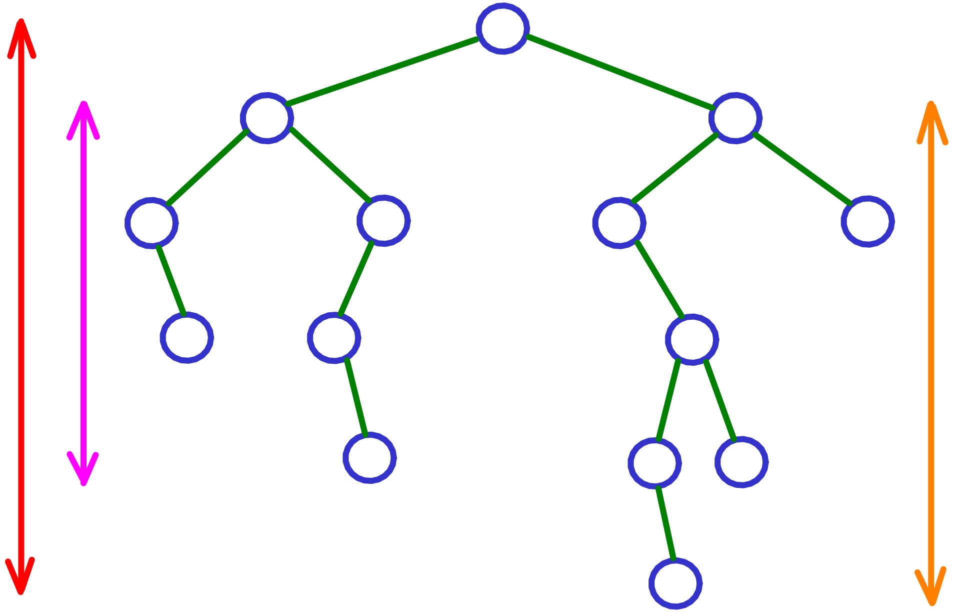
$$= O(\log n)$$ so $E[\text{depth}] \not\Rightarrow$ balance



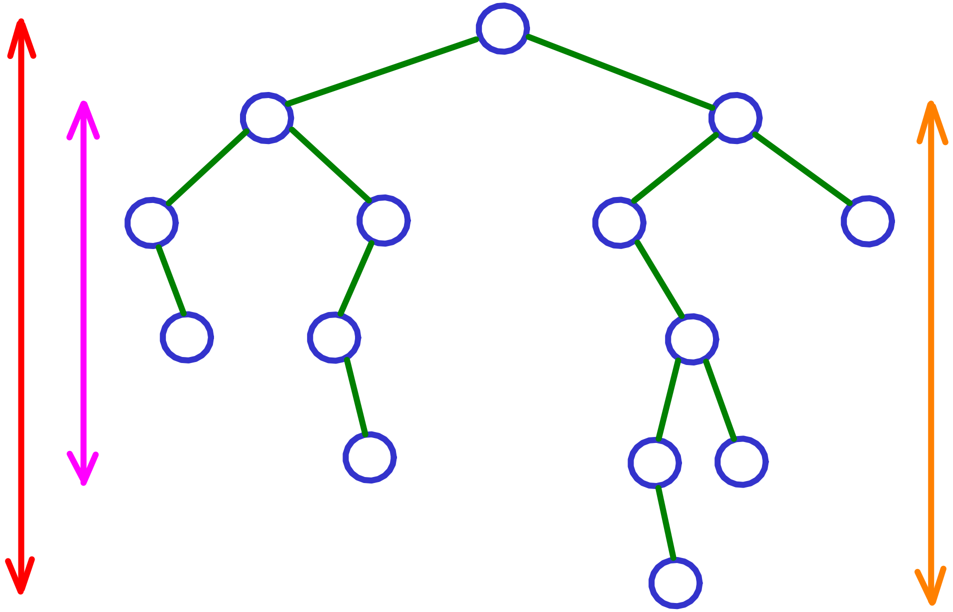
Expected height of randomly built BST



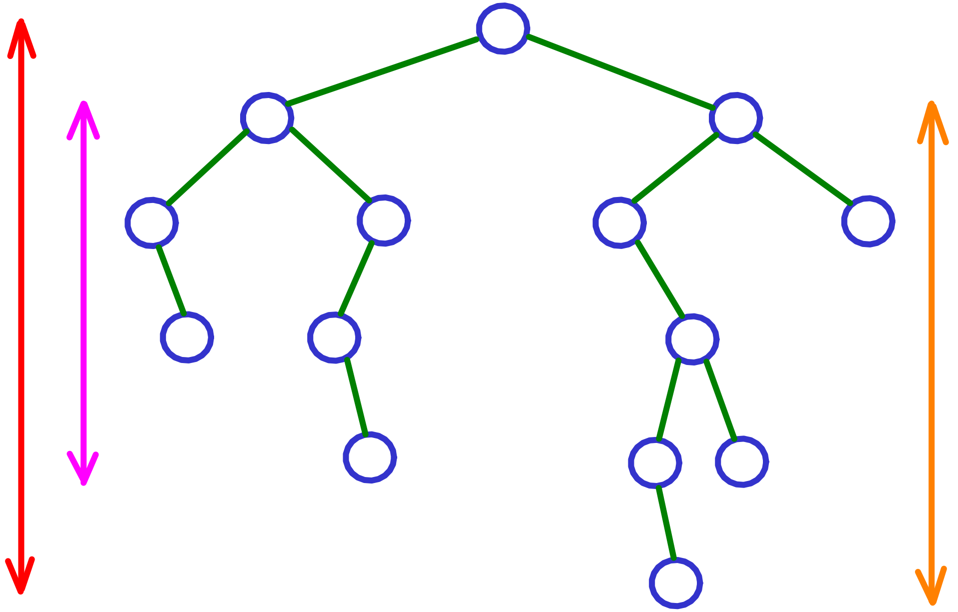
$H(n) = ?$



$H(n) = ?$

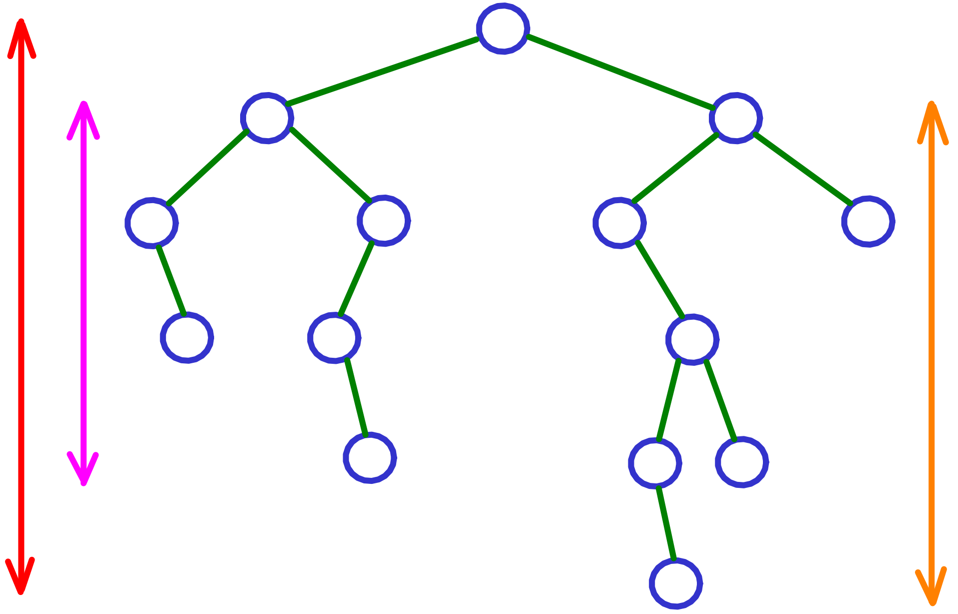


$H(n) = 1 + ?$



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

for some random k $(0 \leq k \leq n-1)$

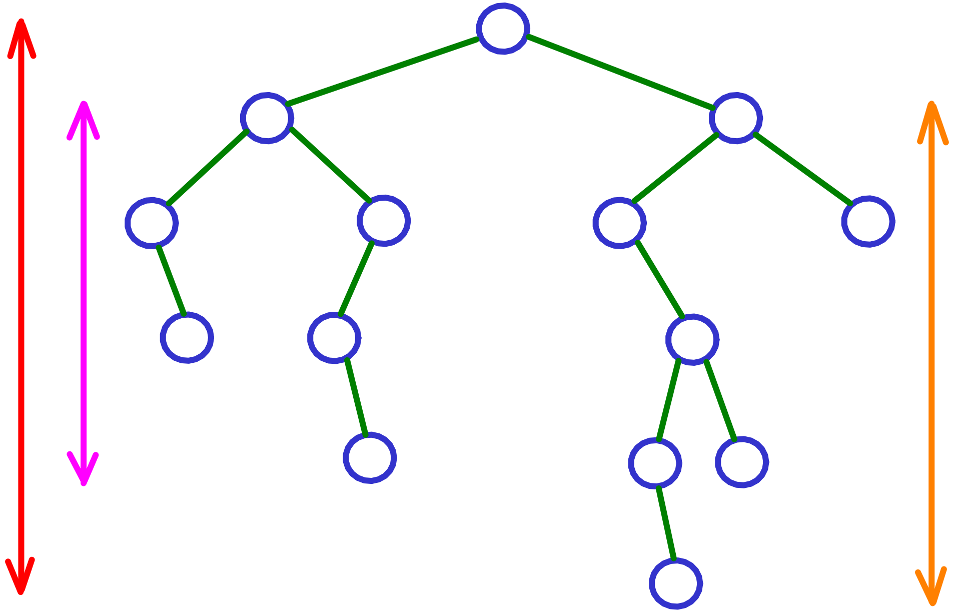


$$H(n) = 1 + \max\{H(k), H(n-k-1)\}$$

for some random k $(0 \leq k \leq n-1)$

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

$\hookrightarrow H(n) \leq$?



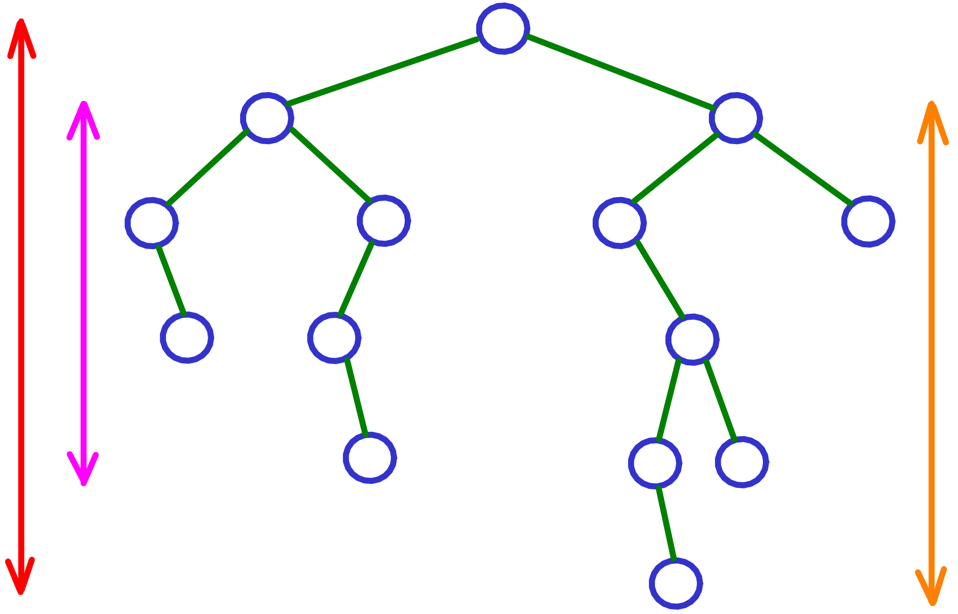
$$H(n) = 1 + \max\{H(k), H(n-k-1)\}$$

for some random k $(0 \leq k \leq n-1)$

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

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

else $H(n) \leq$?



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

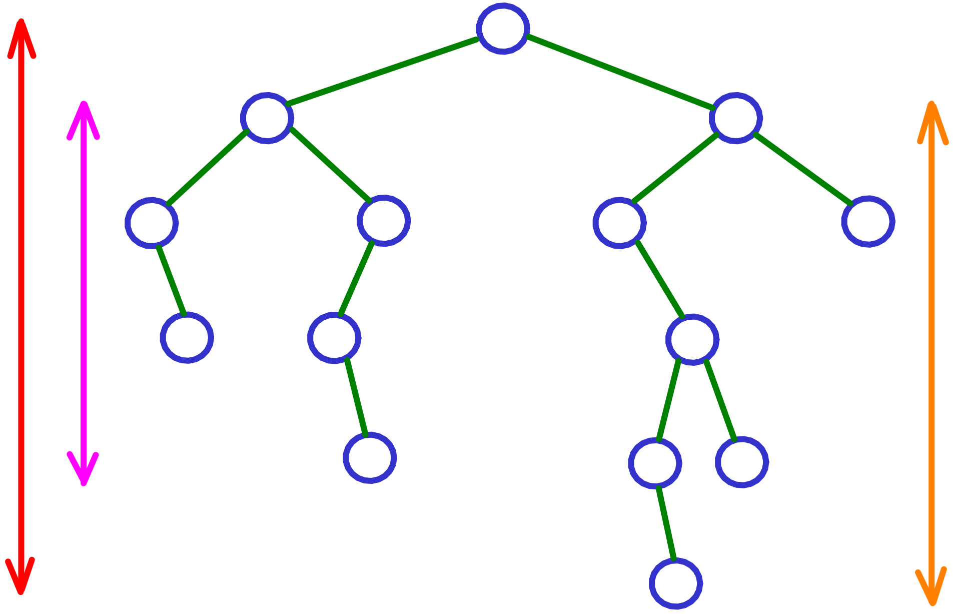
for some random k $(0 \leq k \leq n-1)$

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

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

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

$< 1 + H(n)$



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

for some random k $(0 \leq k \leq n-1)$

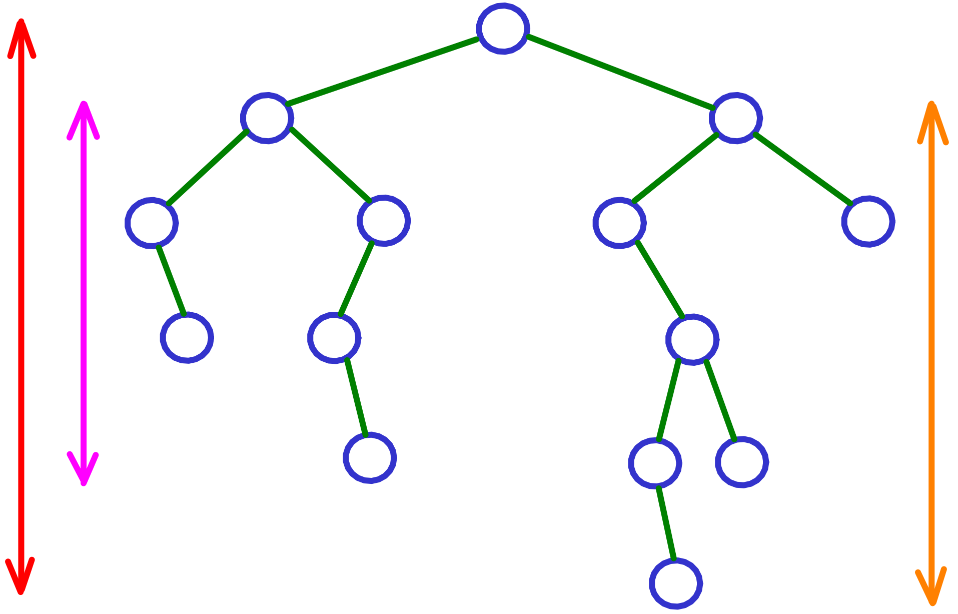
If $\frac{n}{4} < k < \frac{3n}{4}$

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

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

$< 1 + H(n)$

$E[H(n)]$?



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

for some random k $(0 \leq k \leq n-1)$

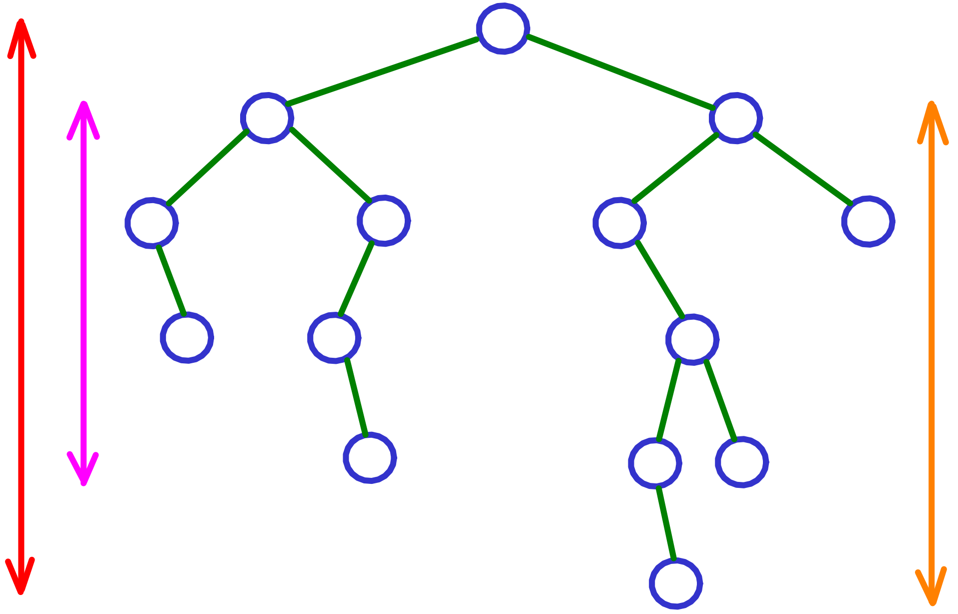
If $\frac{n}{4} < k < \frac{3n}{4}$

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

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

$< 1 + H(n)$

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



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

for some random k $(0 \le k \le n-1)$

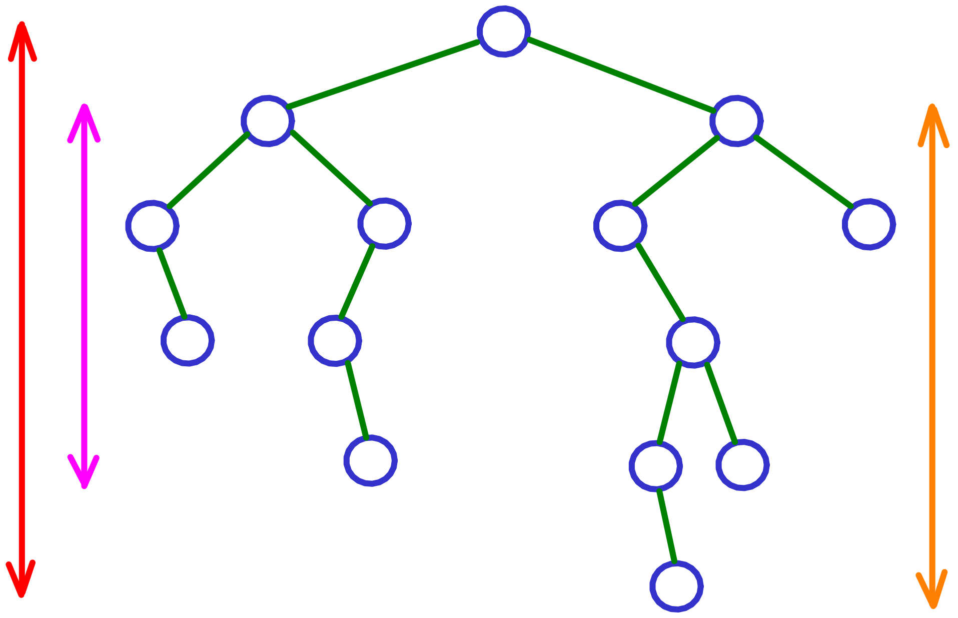
If $\frac{n}{4} < k < \frac{3n}{4}$

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

$$\text{else } H(n) \le 1 + H(n-1) < 1 + H(n)$$

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

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



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

for some random k $(0 \leq k \leq n-1)$

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

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

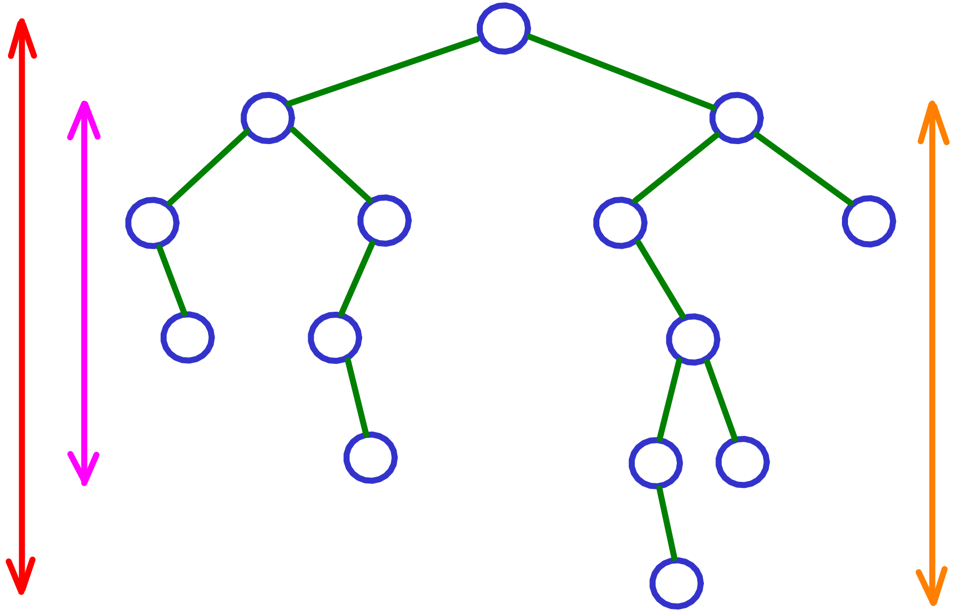
$$\text{else } H(n) \leq 1 + H(n-1)$$

$$< 1 + H(n)$$

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

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

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



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

for some random k $(0 \leq k \leq n-1)$

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

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

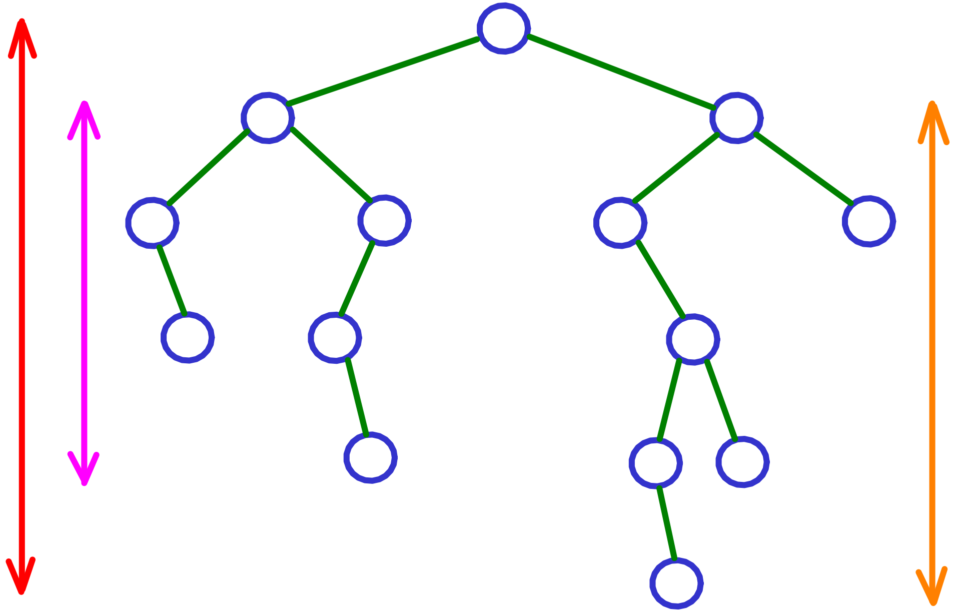
else $H(n) \leq 1 + H(n-1)$

$$< 1 + H(n)$$

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

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

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



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

for some random k $(0 \leq k \leq n-1)$

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

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

$$\text{else } H(n) \leq 1 + H(n-1)$$

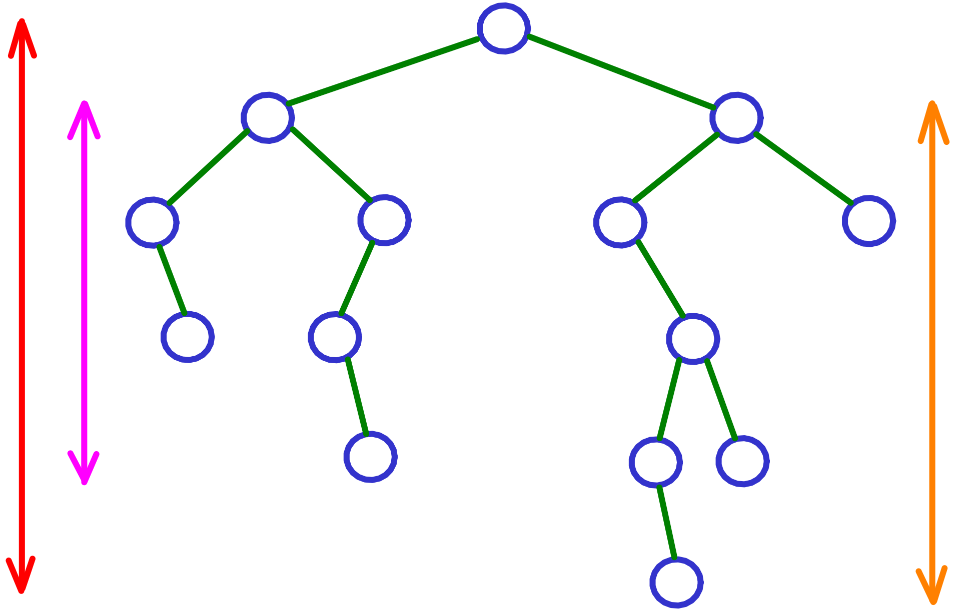
$$< 1 + H(n)$$

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

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

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

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



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

for some random k $(0 \le k \le n-1)$

$\underbrace{\text{random k} \ (0 \le k \le n-1)}$

with rigorous analysis can get $\sim 3\log n$

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

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

$$\text{else } H(n) \le 1 + H(n-1)$$

$$< 1 + H(n)$$

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

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

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

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