

# 1. Insertion into Red-Black Tree

# Insertion into Red-Black Tree

## Task

Consider the sequence

9, 5, 14, 7, 3, 16, 1, 4.

Insert these keys in this order into an empty Red-Black tree and draw each intermediate step of the resulting tree.

# Starting with an empty tree

Initialized (empty) tree

# Starting with an empty tree

Initialized (empty) tree

next up:

# Starting with an empty tree

Initialized (empty) tree

next up:

`insert(9)`

# Inserting 9...

9

`insert(9)`

# Inserting 9...

9

`insert(9)`

next up:

# Inserting 9...

9

`insert(9)`

next up:

`recolor_root()`



# Inserting 9...

9

```
recolor_root()
```

# Inserting 9...

9

`recolor_root()`

next up:

# Inserting 9...

9

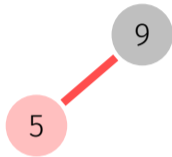
`recolor_root()`

next up:

`insert(5)`

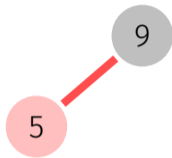
**Case:** Parent node  $x$  is  
black with  $k < x$

# Inserting 5...



`insert(5)`

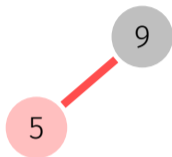
# Inserting 5...



`insert(5)`

next up:

# Inserting 5...



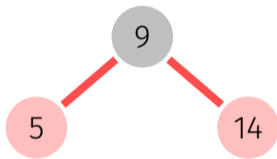
`insert(5)`

next up:

`insert(14)`

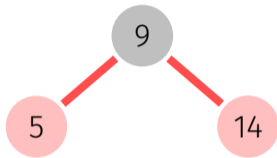
**Case:** Parent node  $y$  is black with  $y < k$  and has a red left child  $x$

# Inserting 14...



`insert(14)`

# Inserting 14...

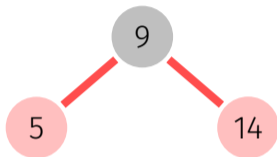


`insert(14)`

next up:



# Inserting 14...

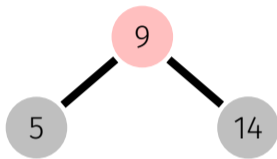


`insert(14)`

next up:

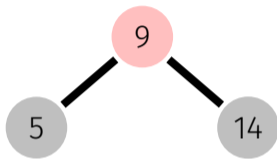
`color_flip(9)`

# Inserting 14...



`color_flip(9)`

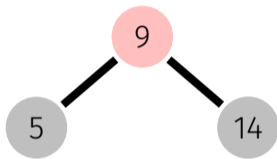
# Inserting 14...



`color_flip(9)`

next up:

# Inserting 14...

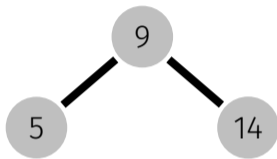


`color_flip(9)`

next up:

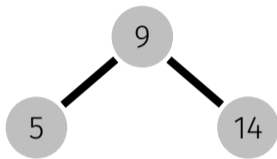
`recolor_root()`

# Inserting 14...



`recolor_root()`

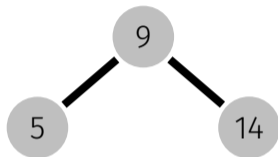
# Inserting 14...



`recolor_root()`

next up:

# Inserting 14...



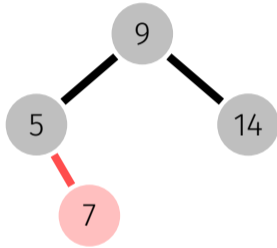
`recolor_root()`

next up:

`insert(7)`

**Case:** Parent node  $x$  is  
black with  $x < k$

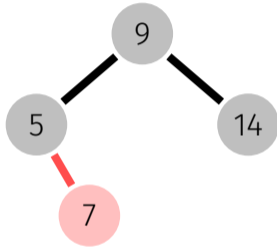
# Inserting 7...



`insert(7)`



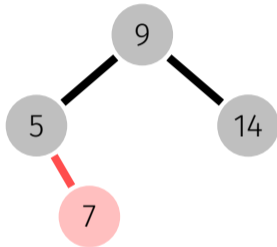
# Inserting 7...



`insert(7)`

next up:

# Inserting 7...

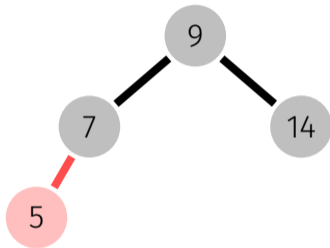


`insert(7)`

next up:

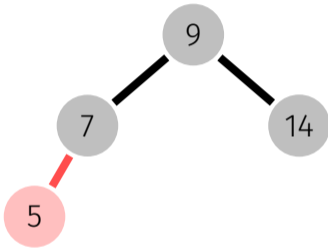
`left_rotation(7)`

# Inserting 7...



`left_rotation(7)`

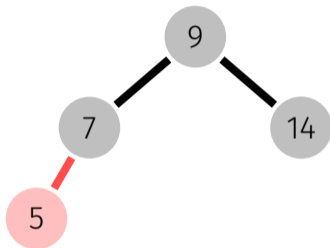
# Inserting 7...



`left_rotation(7)`

next up:

# Inserting 7...



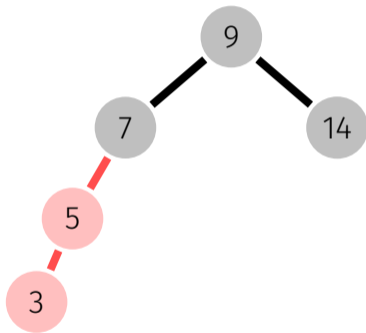
`left_rotation(7)`

next up:

`insert(3)`

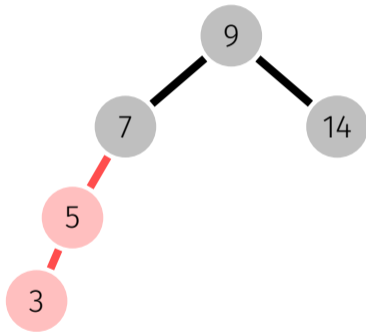
**Case:** Parent node  $x$  is red  
with  $k < x$

# Inserting 3...



`insert(3)`

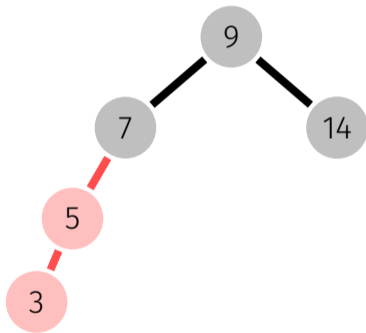
# Inserting 3...



`insert(3)`

next up:

# Inserting 3...



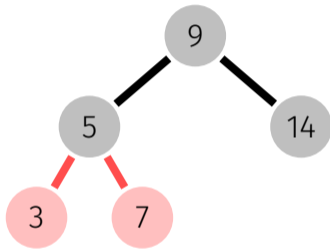
`insert(3)`

next up:

`right_rotation(5)`

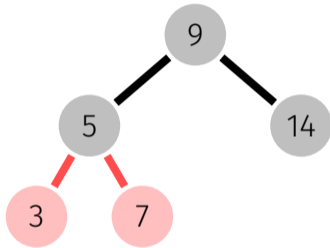


# Inserting 3...



`right_rotation(5)`

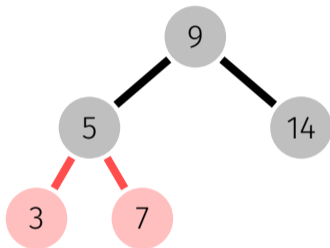
# Inserting 3...



`right_rotation(5)`

next up:

# Inserting 3...

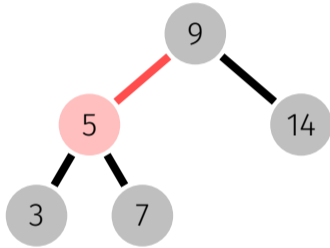


`right_rotation(5)`

next up:

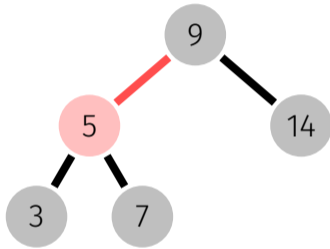
`color_flip(5)`

# Inserting 3...



`color_flip(5)`

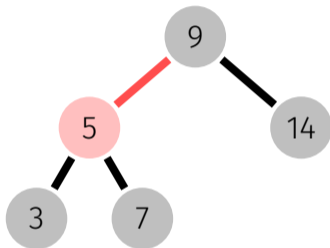
# Inserting 3...



`color_flip(5)`

next up:

# Inserting 3...



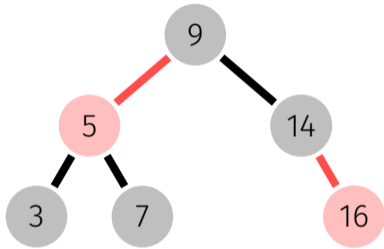
`color_flip(5)`

next up:

`insert(16)`

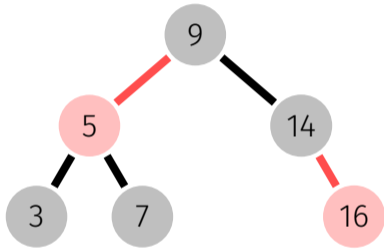
**Case:** Parent node  $x$  is black with  $x < k$

# Inserting 16...



`insert(16)`

# Inserting 16...

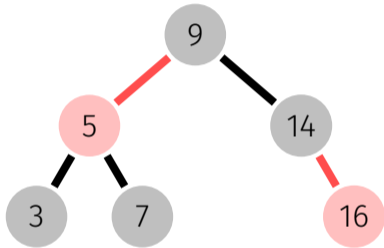


`insert(16)`

next up:



# Inserting 16...

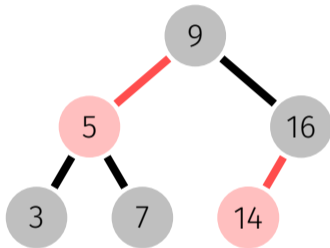


`insert(16)`

next up:

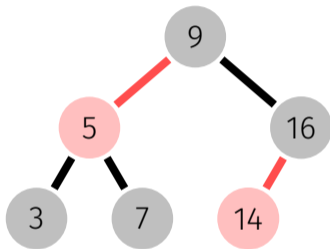
`left_rotation(16)`

# Inserting 16...



`left_rotation(16)`

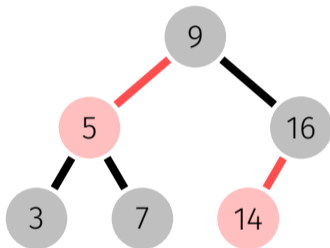
# Inserting 16...



`left_rotation(16)`

next up:

# Inserting 16...



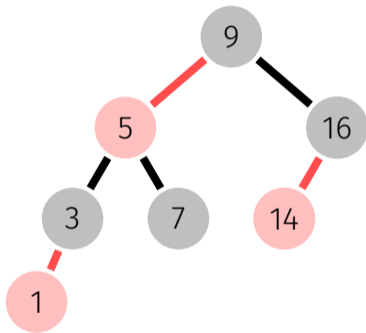
`left_rotation(16)`

next up:

`insert(1)`

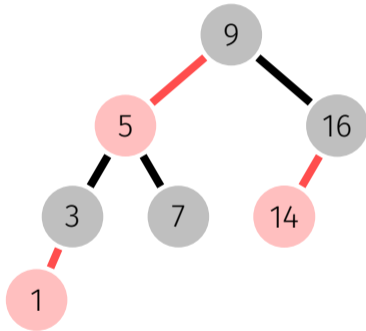
**Case:** Parent node  $x$  is black with  $k < x$

# Inserting 1...



`insert(1)`

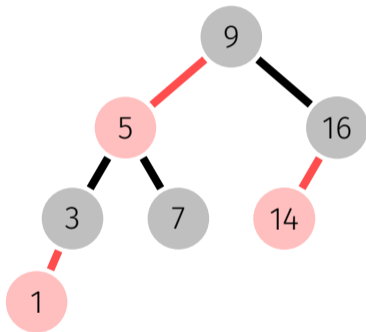
# Inserting 1...



`insert(1)`

next up:

# Inserting 1...



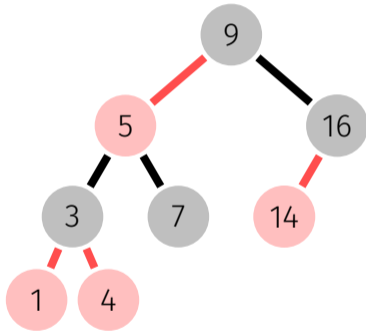
`insert(1)`

next up:

`insert(4)`

**Case:** Parent node  $y$  is black with  $y < k$  and  $y$  has red left child  $x$

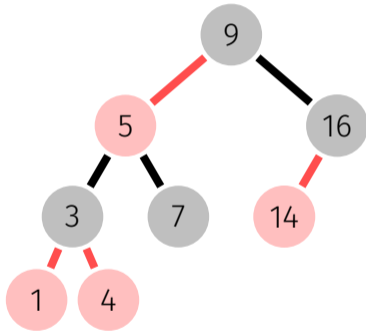
# Inserting 4...



`insert(4)`



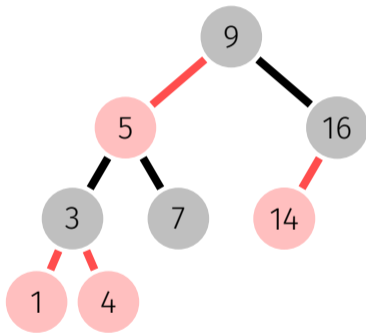
# Inserting 4...



`insert(4)`

next up:

# Inserting 4...

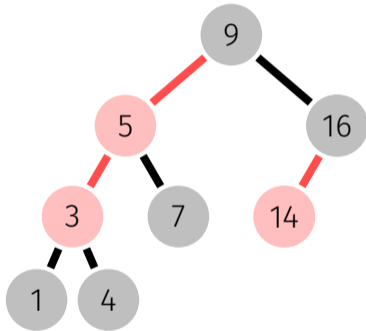


`insert(4)`

next up:

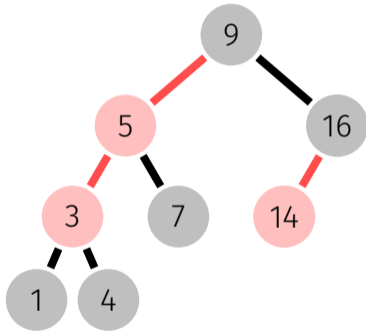
`color_flip(3)`

# Inserting 4...



`color_flip(3)`

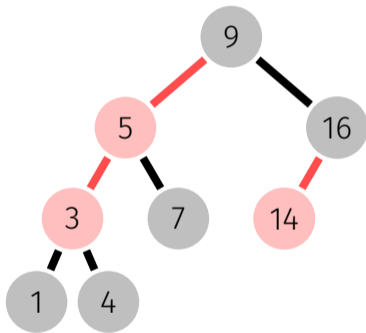
# Inserting 4...



`color_flip(3)`

next up:

# Inserting 4...

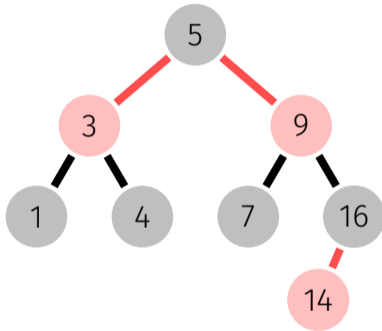


`color_flip(3)`

next up:

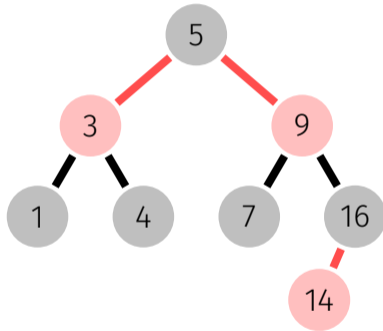
`right_rotation(5)`

# Inserting 4...



`right_rotation(5)`

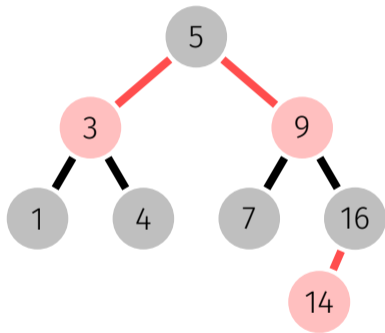
# Inserting 4...



`right_rotation(5)`

next up:

# Inserting 4...



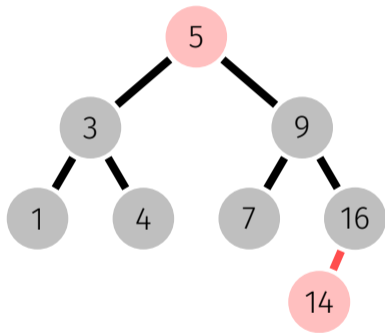
`right_rotation(5)`

next up:

`color_flip(5)`

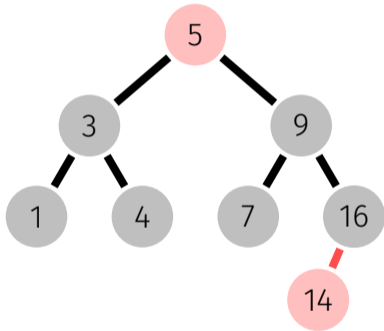


# Inserting 4...



`color_flip(5)`

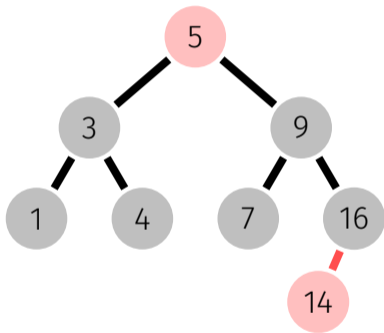
# Inserting 4...



`color_flip(5)`

next up:

# Inserting 4...

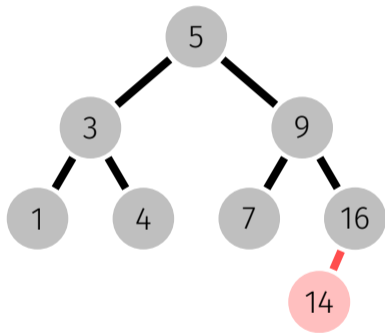


`color_flip(5)`

next up:

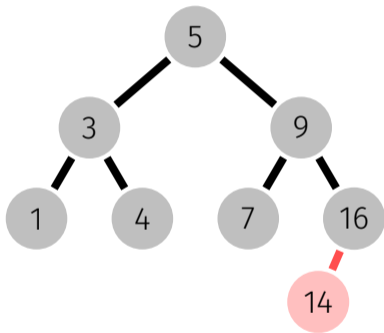
`recolor_root()`

# Done!



`recolor_root()`

# Done!



## Pre-order

5 3 1 4 9 7 16 14

## Post-order

1 4 3 7 14 16 9 5

## In-order

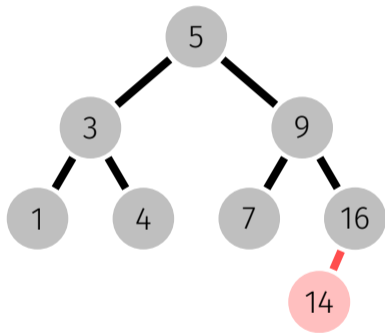
1 3 4 5 7 9 14 16

# And another one!

## **Task**

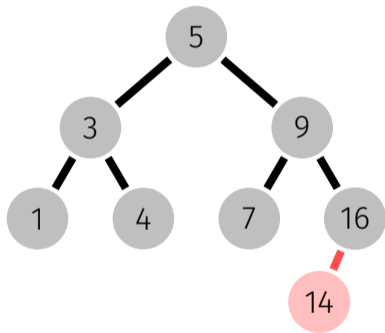
Now insert the key 2 and draw the tree again.

## Inserting 2...



Start with last tree

# Inserting 2...

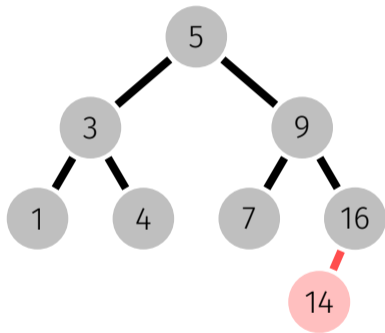


Start with last tree

next up:



# Inserting 2...



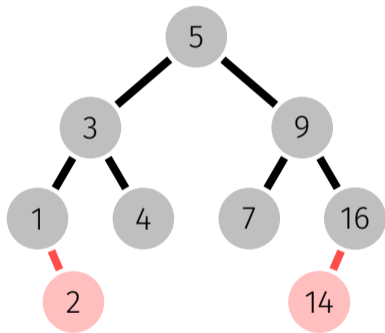
Start with last tree

next up:

`insert(2)`

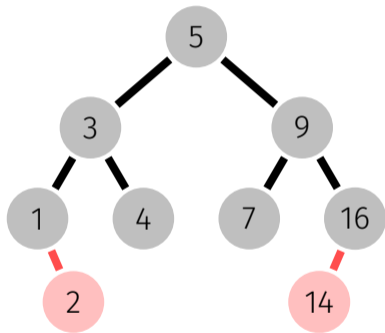
**Case:** Parent node  $x$  is black with  $x < k$

# Inserting 2...



`insert(2)`

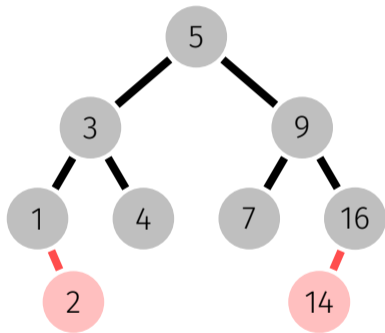
# Inserting 2...



`insert(2)`

next up:

# Inserting 2...

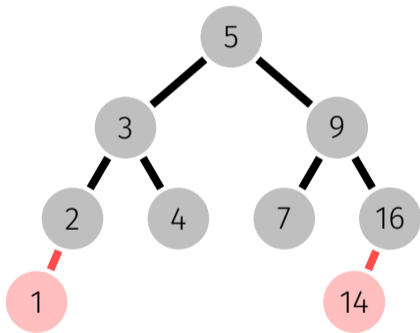


`insert(2)`

next up:

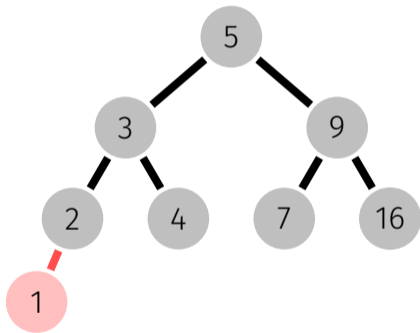
`left_rotation(2)`

# 2 inserted!



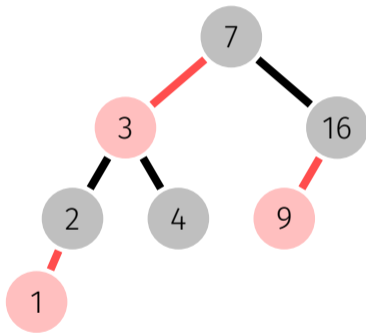
`left_rotation(2)`

(14 removed!)



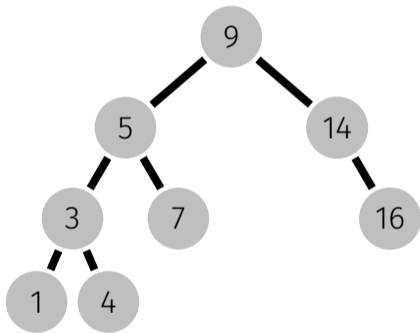
(optional task)

(5 removed!)



(optional task)

# Other solutions: Filled Binary Search Tree



## Pre-order

9 5 3 1 4 7 14 16

## Post-order

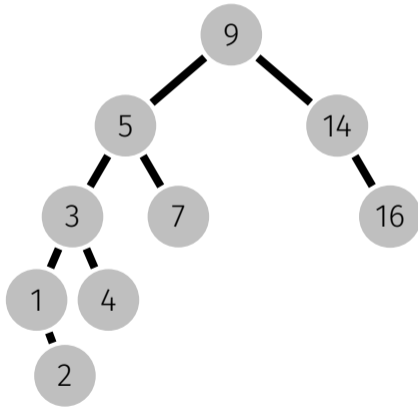
1 4 3 7 5 16 14 9

## In-order

1 3 4 5 7 9 14 16

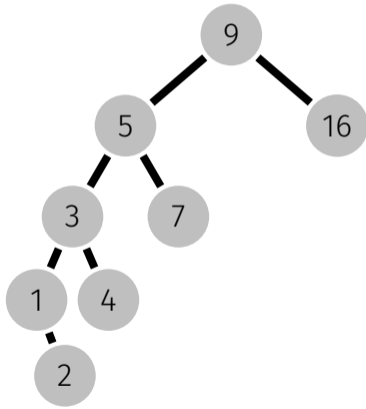


# Other solutions: Filled Binary Search Tree



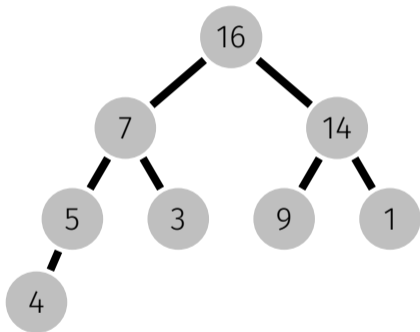
After inserting 2

# Other solutions: Filled Binary Search Tree



After removing 14

# Other solutions: Filled Max-Heap



## Pre-order

16 7 5 4 3 14 9 1

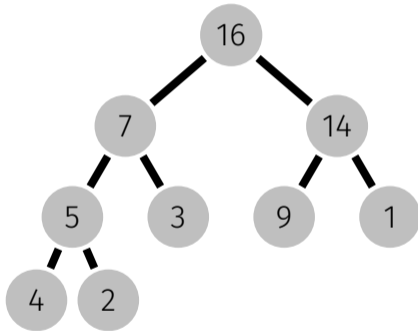
## Post-order

4 5 3 7 9 1 14 16

## In-order

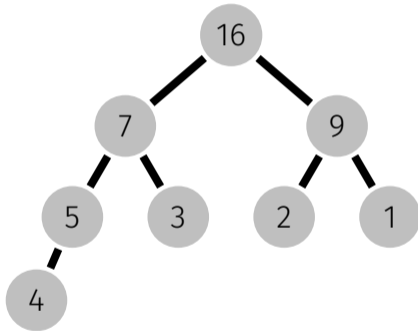
4 5 7 3 16 9 14 1

# Other solutions: Filled Max-Heap



After inserting 2

# Other solutions: Filled Max-Heap



After removing 14