Algorithms

FOURTH EDITION

ROBERT SEDGEWICK │ KEVIN WAYNE

http://algs4.cs.princeton.edu

3.3 2-3 TREE DEMO

‣ *search*

‣ *insertion*
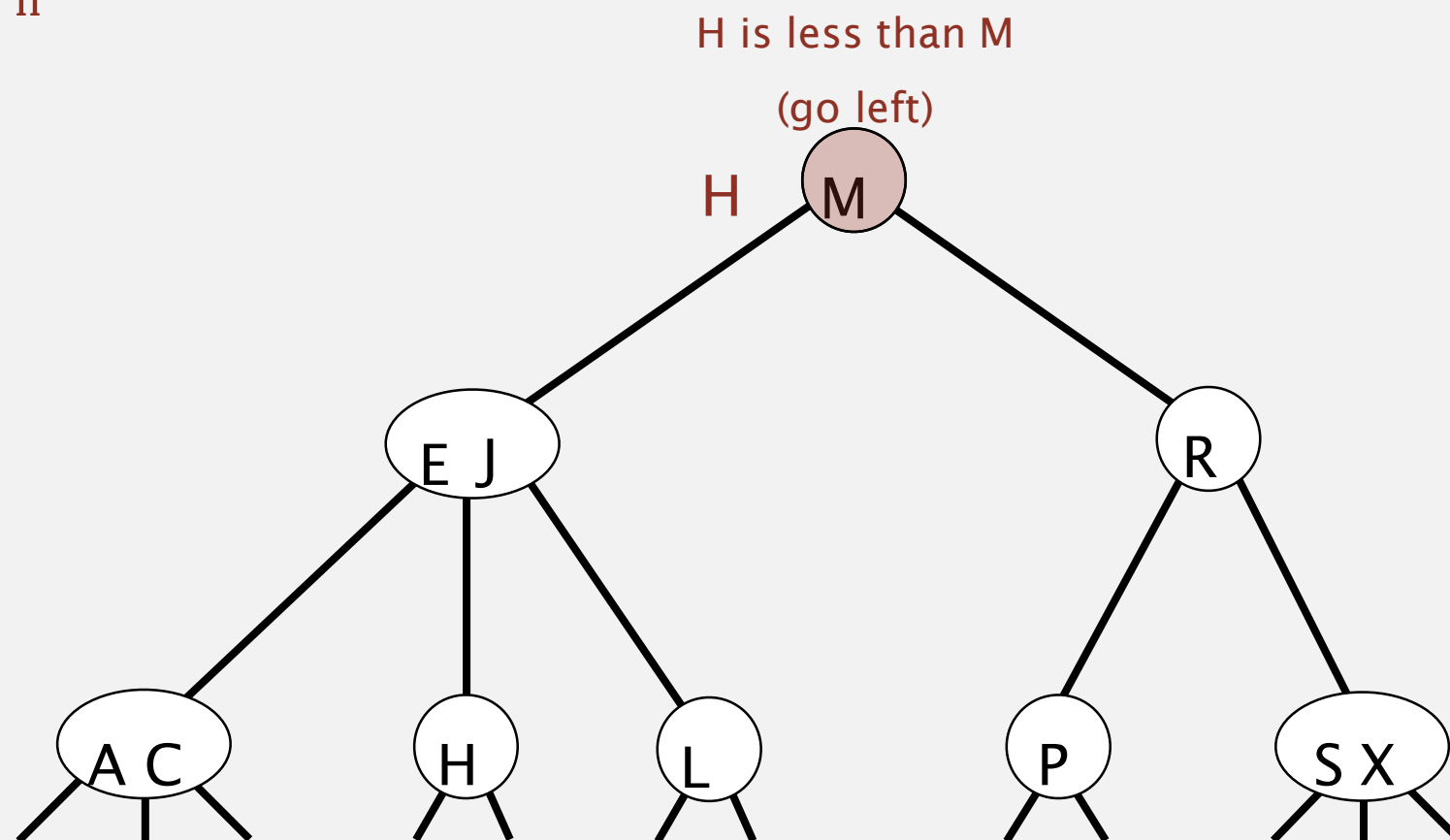
‣ *construction*

3.3 2–3 Tree Demo

‣ *search*

‣ *insertion*

‣ *construction*

Search.

- Compare search key against keys in node.
- Find interval containing search key.
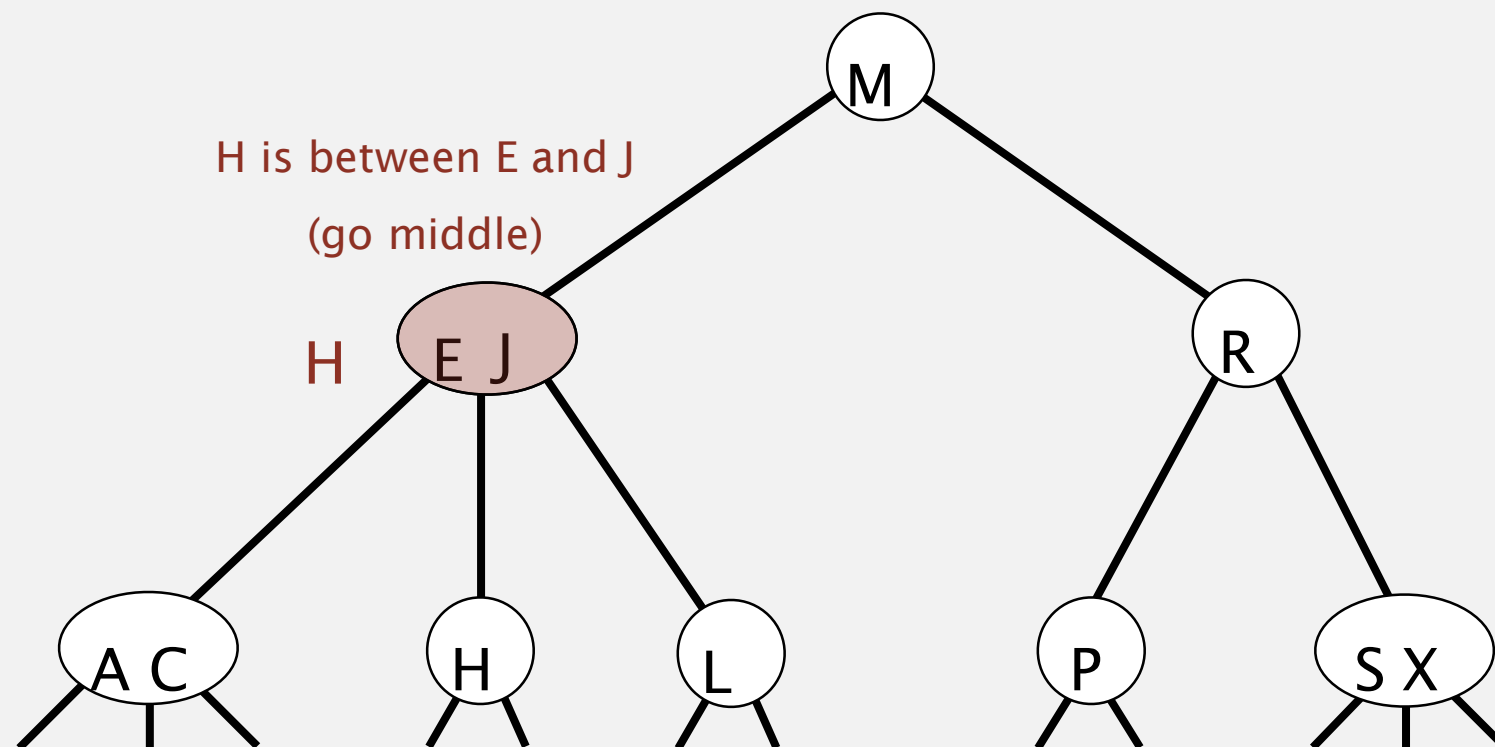- Follow associated link (recursively).

search for H

H is less than M

(go left)

H M

E J

R

A C

H

L

P

S X

Search.

- Compare search key against keys in node.
- Find interval containing search key.
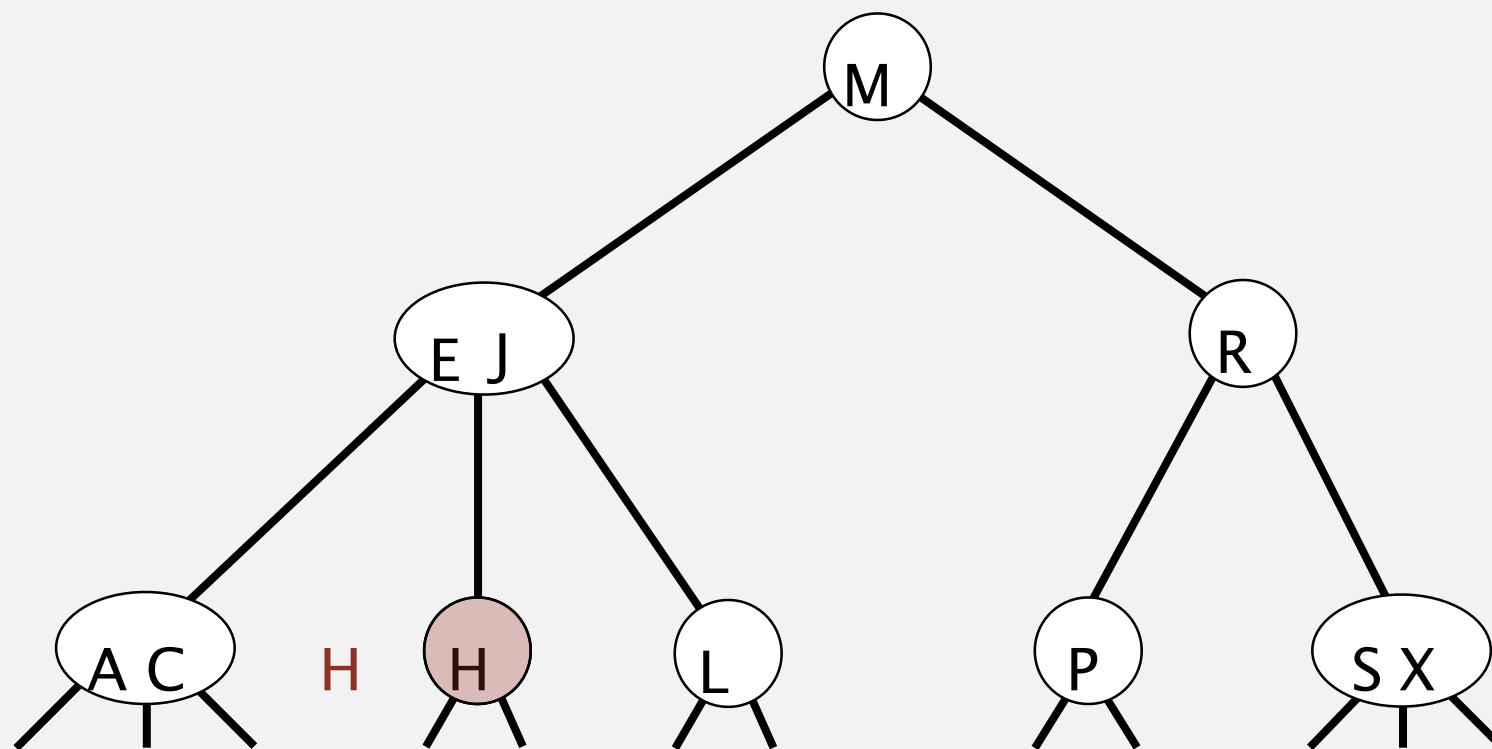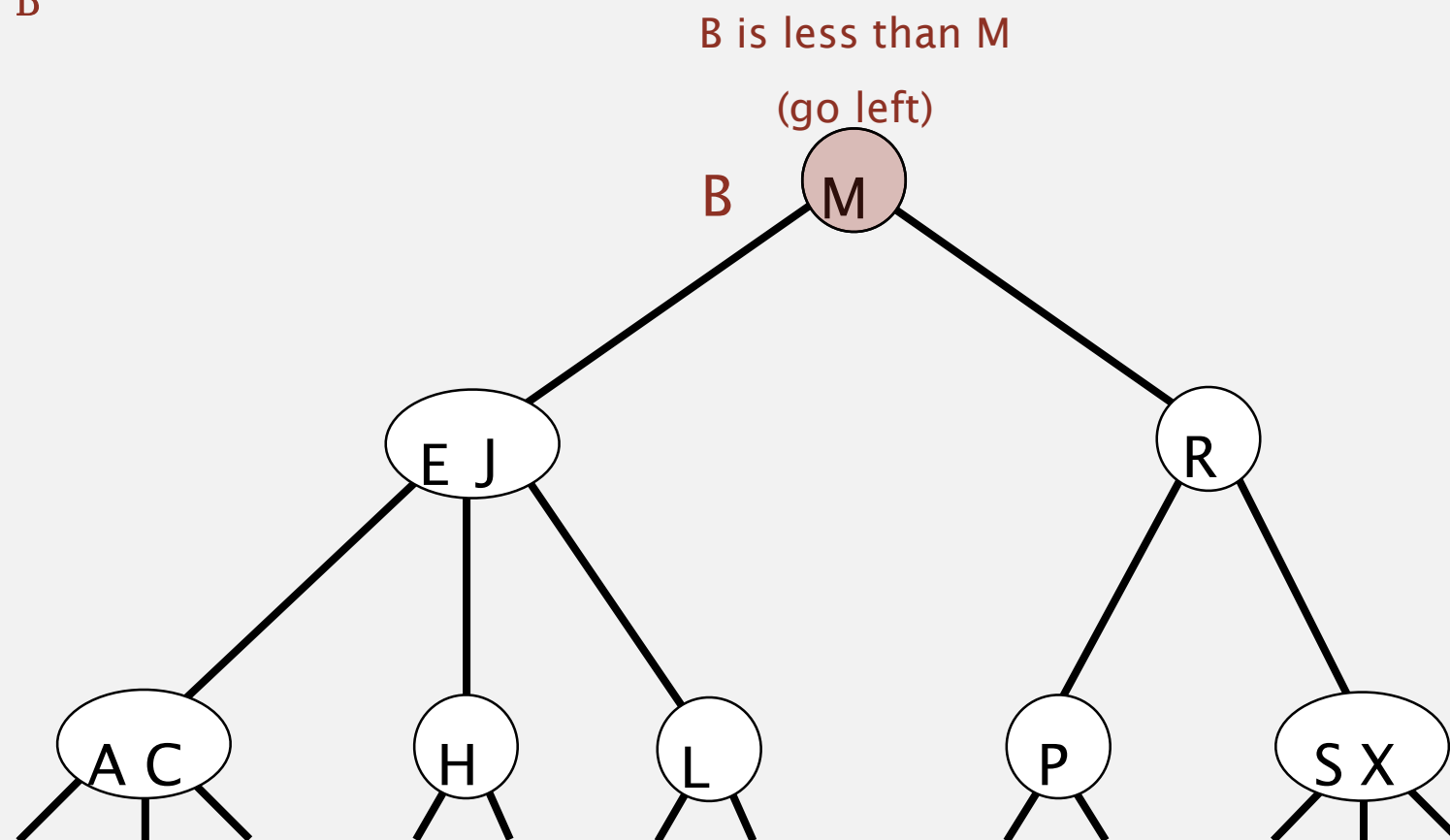- Follow associated link (recursively).

search for H

H is between E and J
(go middle)

Search.

- Compare search key against keys in node.
- Find interval containing search key.
- Follow associated link (recursively).

search for H

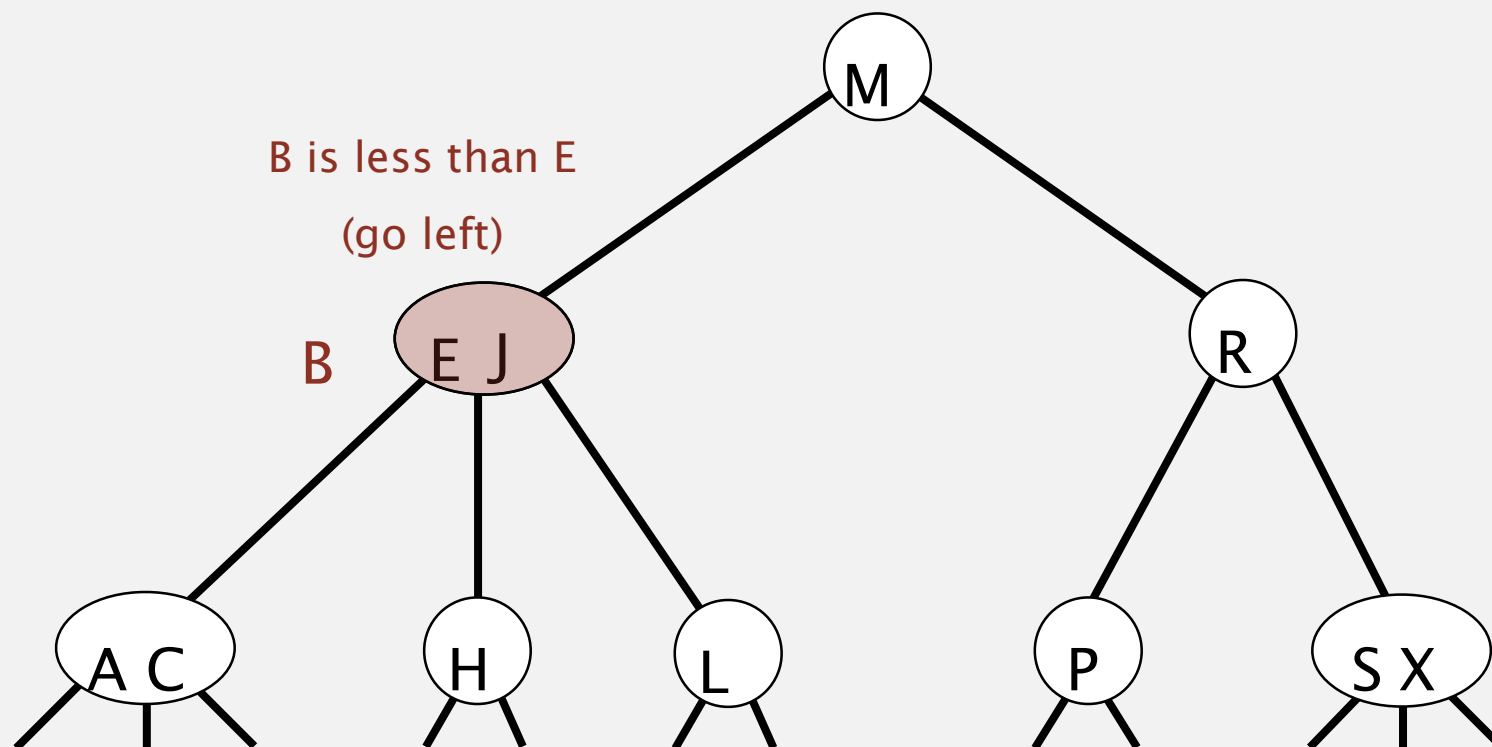found H

(search hit)

Search.

- Compare search key against keys in node.
- Find interval containing search key.
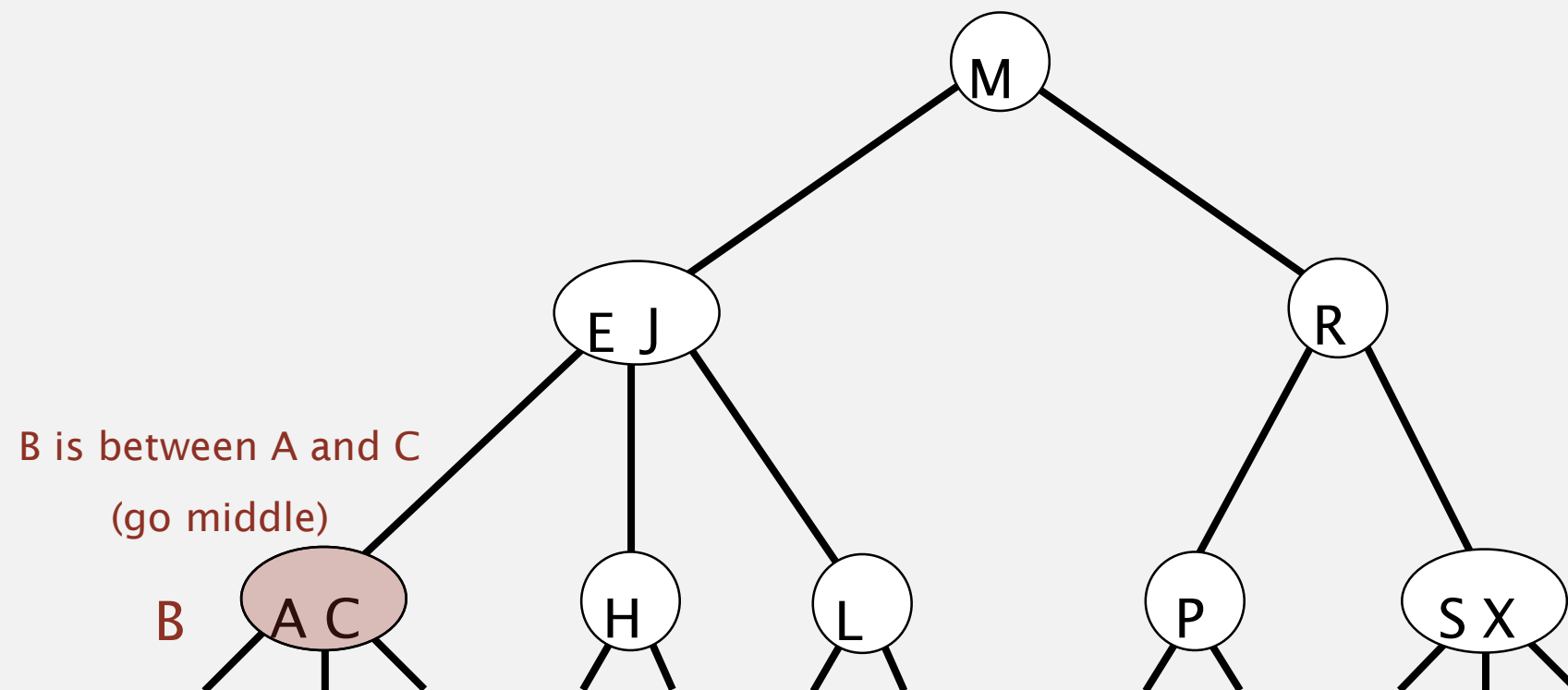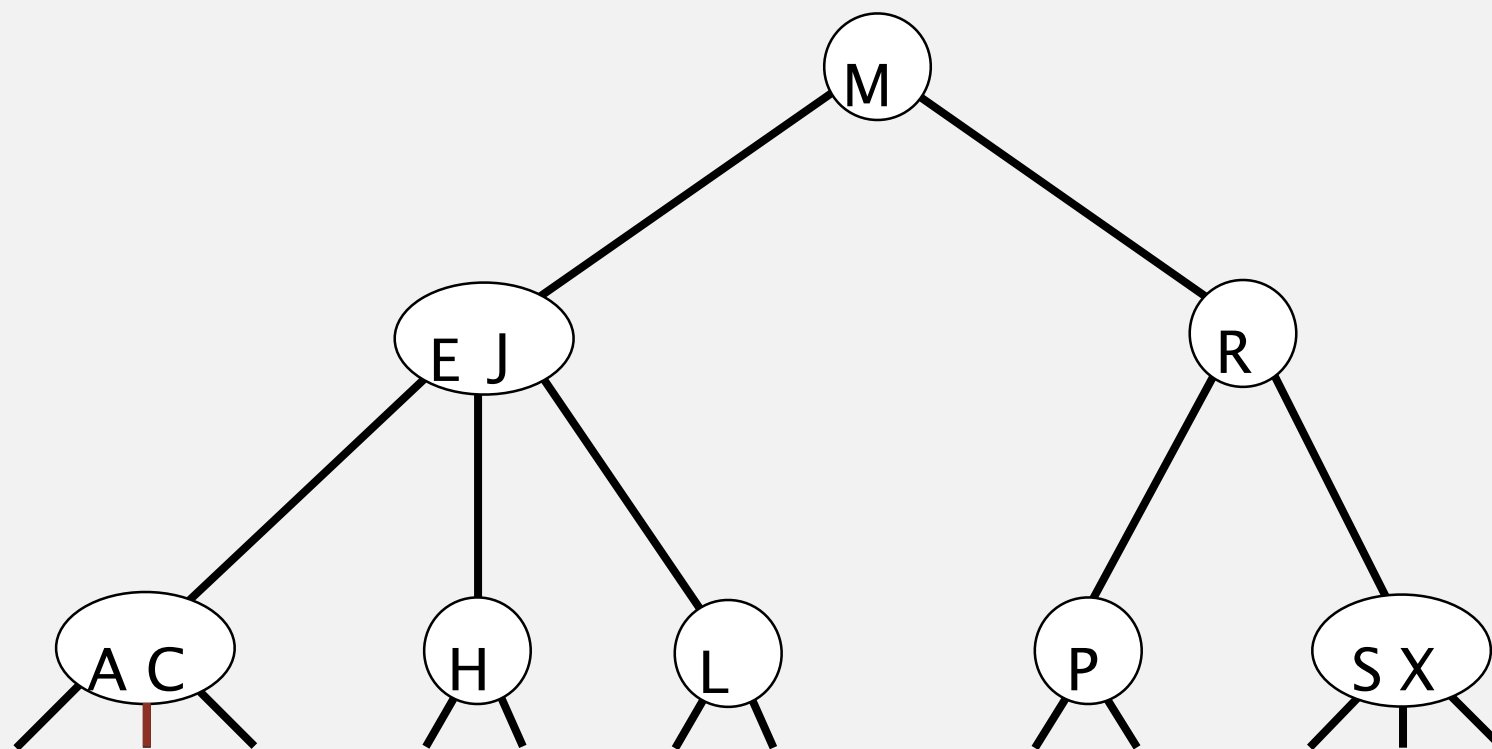- Follow associated link (recursively).

search for B

B is less than M

(go left)

B   M

Search.

- Compare search key against keys in node.
- Find interval containing search key.
- Follow associated link (recursively).

search for B

Search.

- Compare search key against keys in node.
- Find interval containing search key.
- Follow associated link (recursively).

search for B

B is between A and C
(go middle)

Search.

- Compare search key against keys in node.
- Find interval containing search key.
- Follow associated link (recursively).

search for B

M

E J

R

A C

H

L

P

S X

B

link is null

(search miss)

Algorithms

Robert Sedgewick | Kevin Wayne

3.3 2-3 Tree Demo

‣ *search*

‣ *insertion*

‣ *construction*

Insert into a 2-node at bottom.
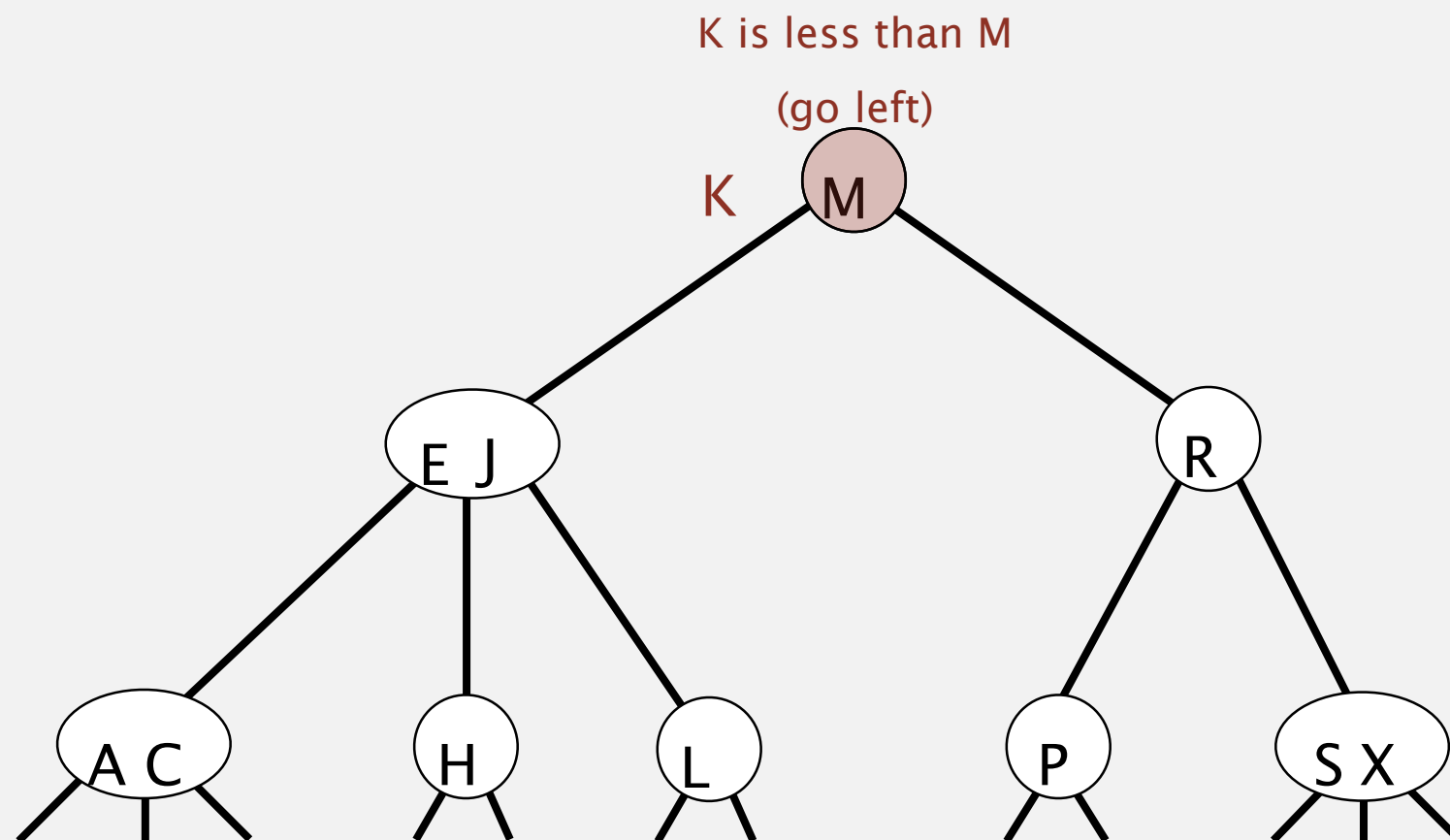
- Search for key, as usual.
- Replace 2-node with 3-node.

insert K

K is less than M

(go left)

K   M

E J   R

A C   H   L   P   S X

Insert into a 2-node at bottom.

- Search for key, as usual.
- Replace 2-node with 3-node.

insert K

K is greater than J
(go right)
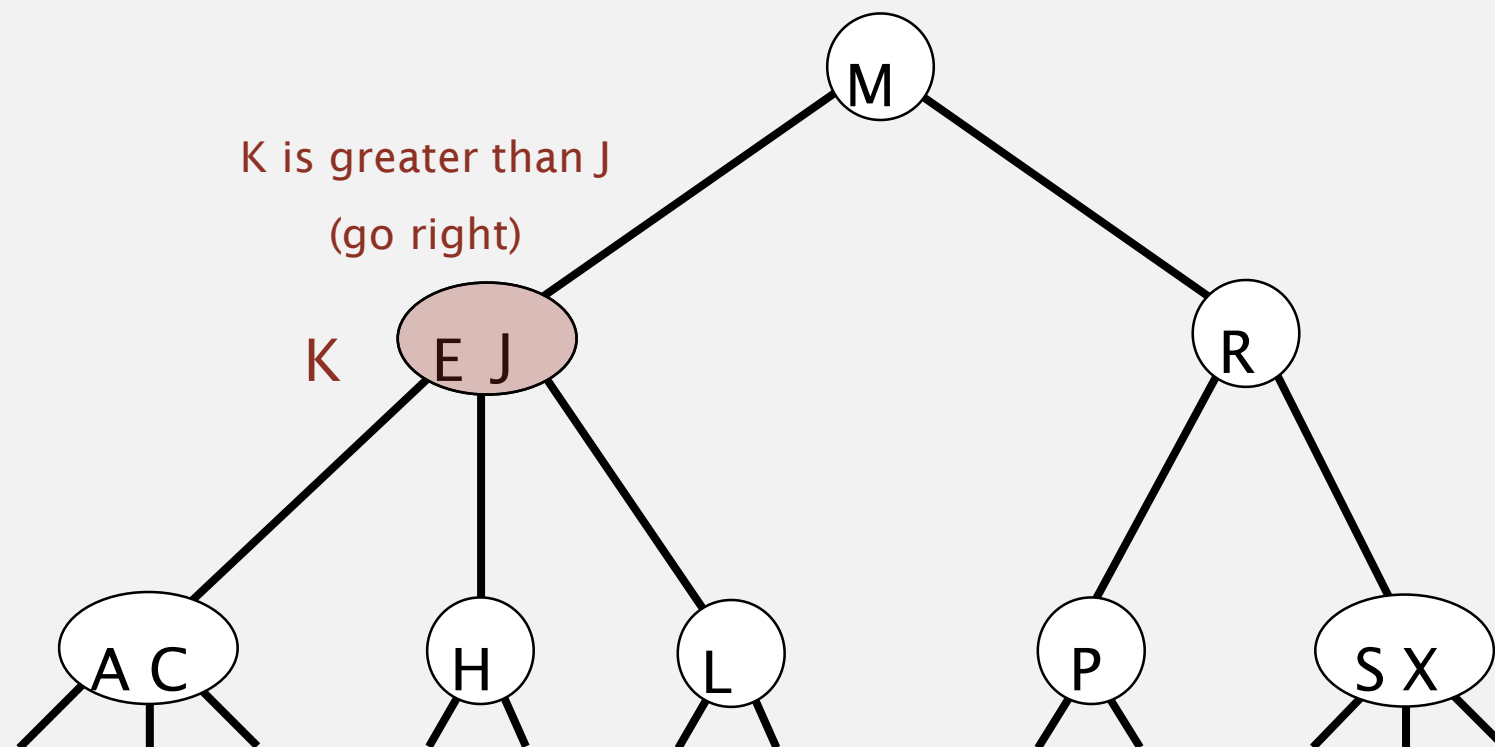
Insert into a 2-node at bottom.
- Search for key, as usual.
- Replace 2-node with 3-node.
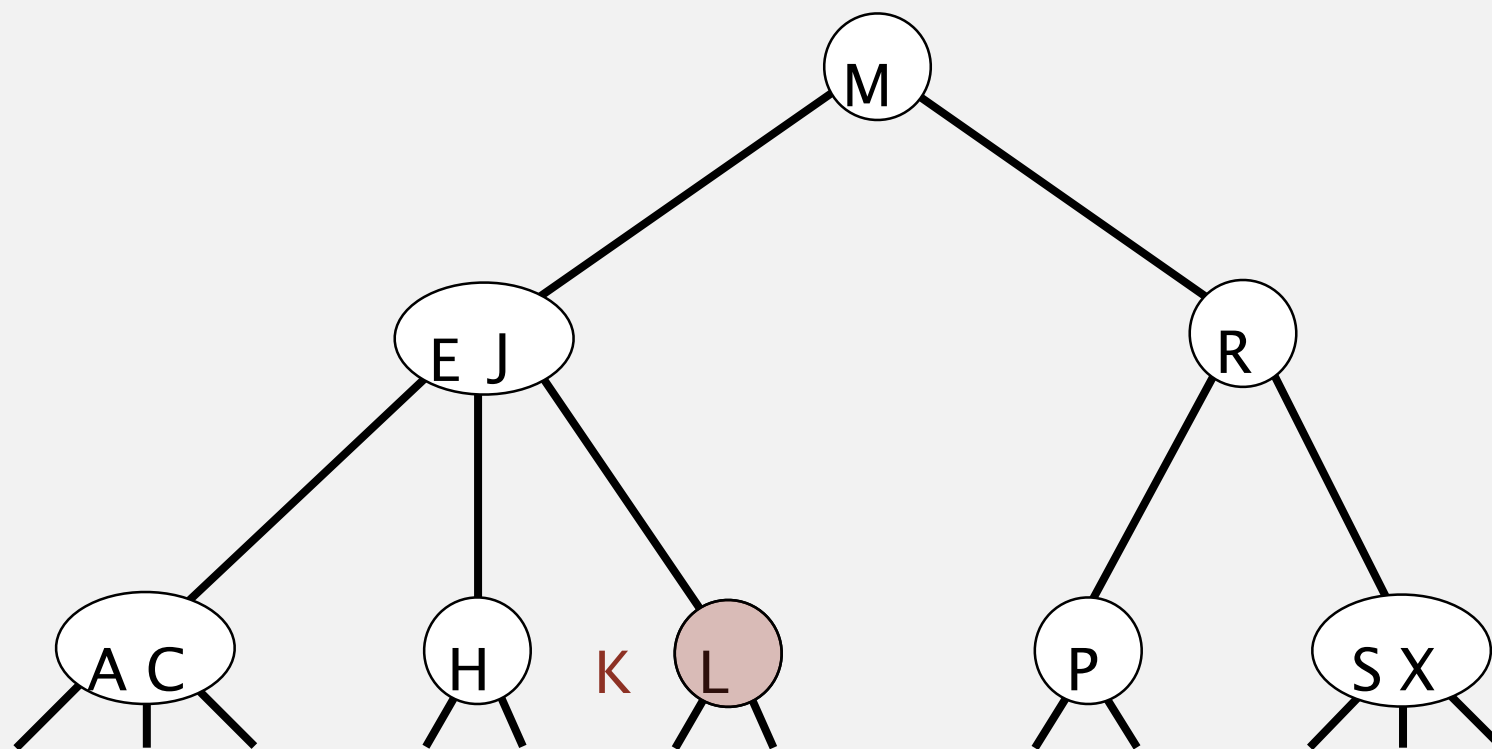
insert K

search ends here

Insert into a 2-node at bottom.
- Search for key, as usual.
- Replace 2-node with 3-node.
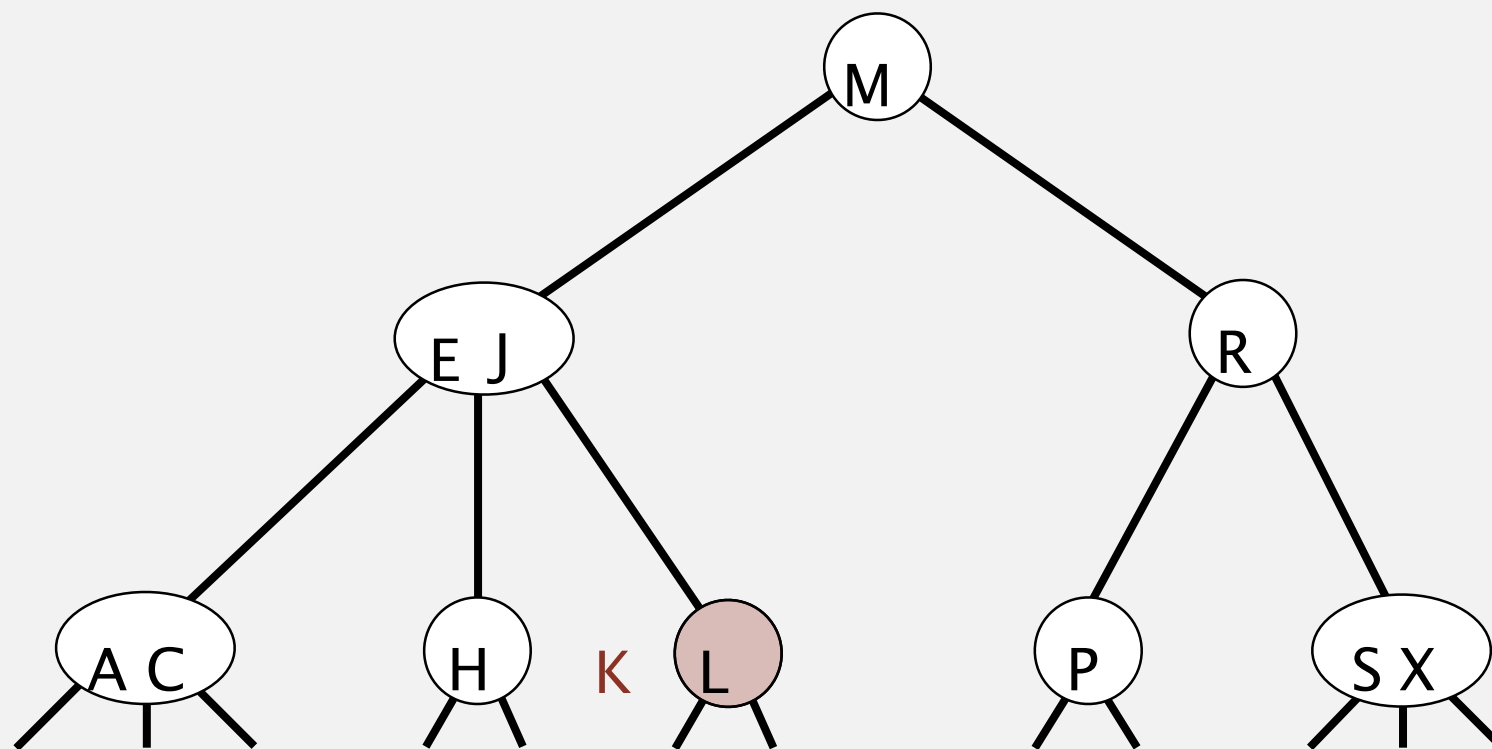
insert K

replace 2-node with

3-node containing K

Insert into a 2-node at bottom.

- Search for key, as usual.
- Replace 2-node with 3-node.

insert K

Insert into a 3-node at bottom.

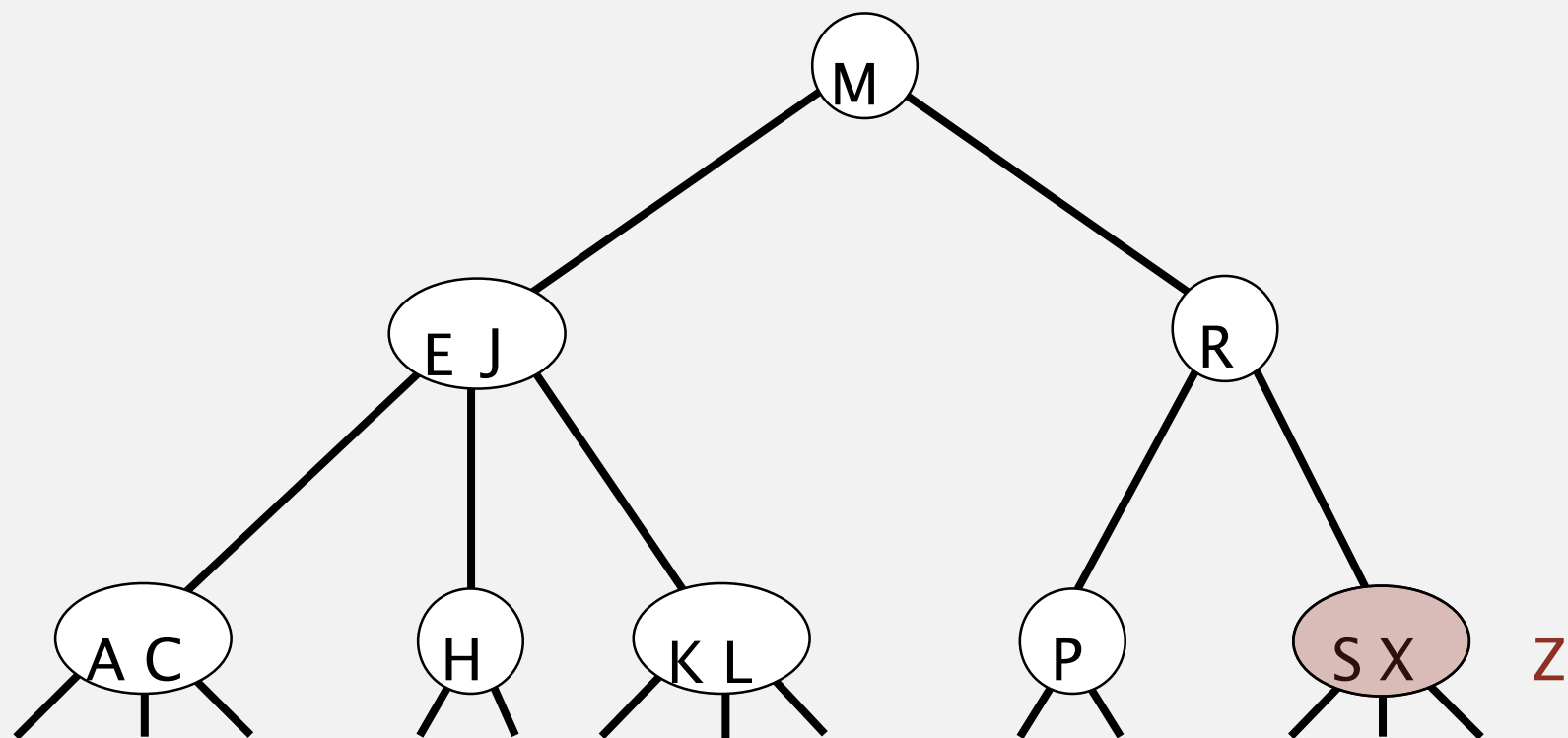- Add new key to 3-node to create temporary 4-node.
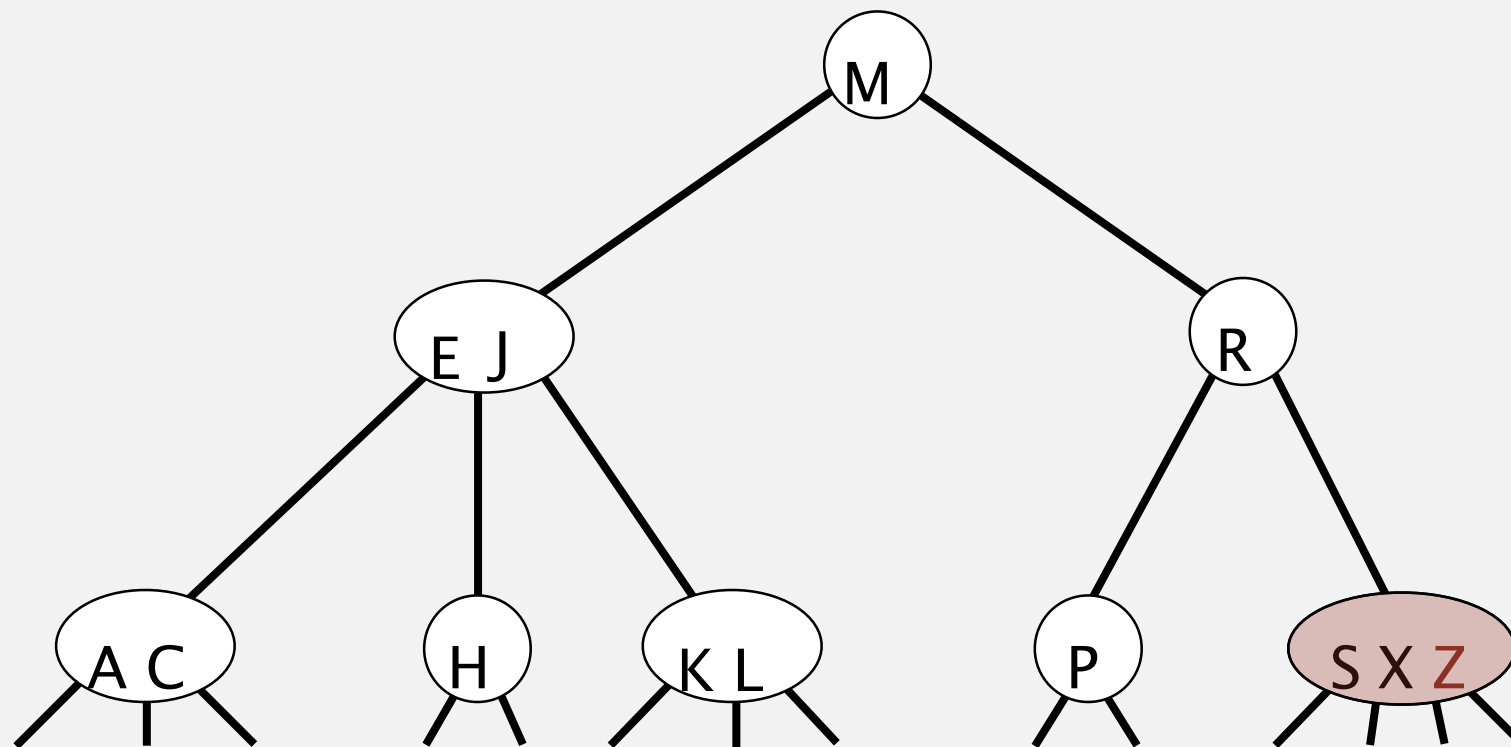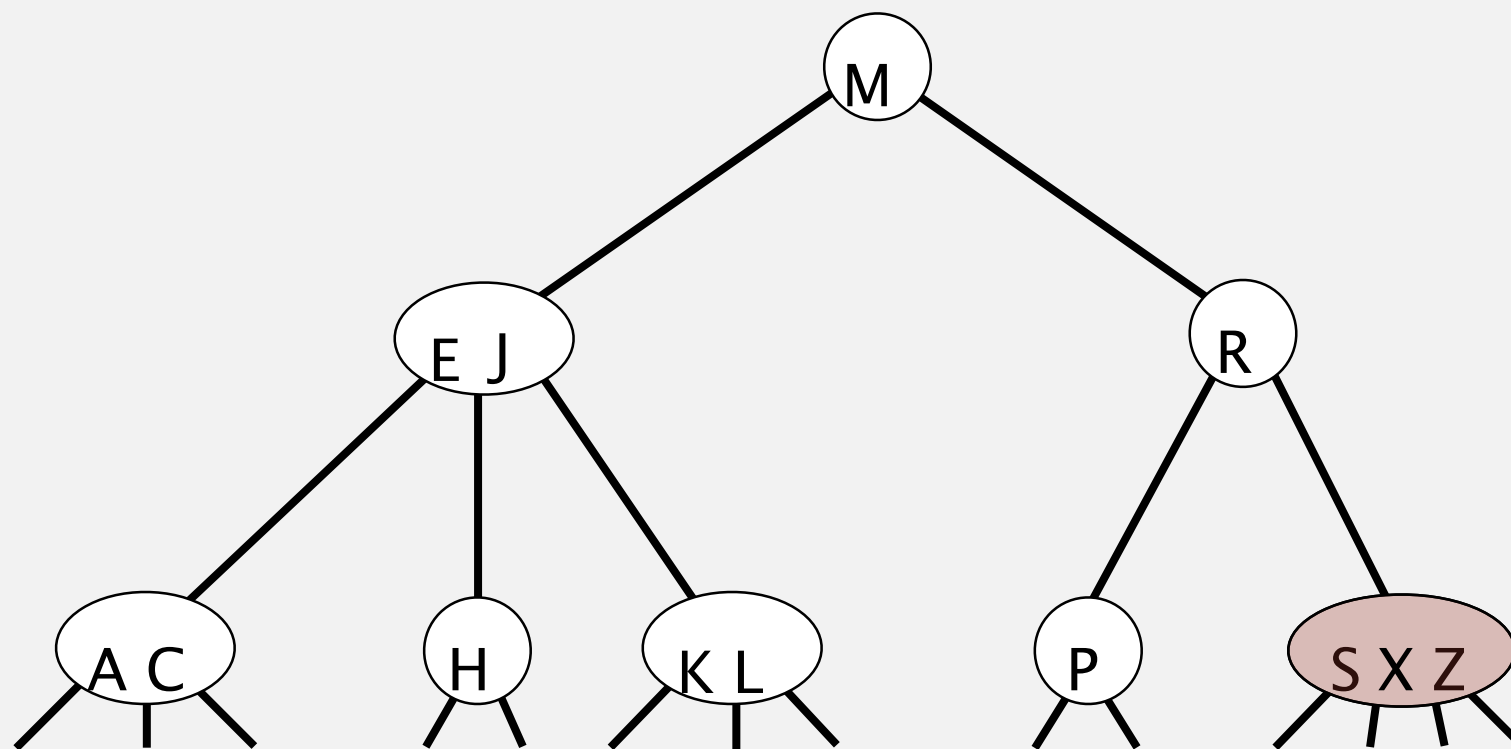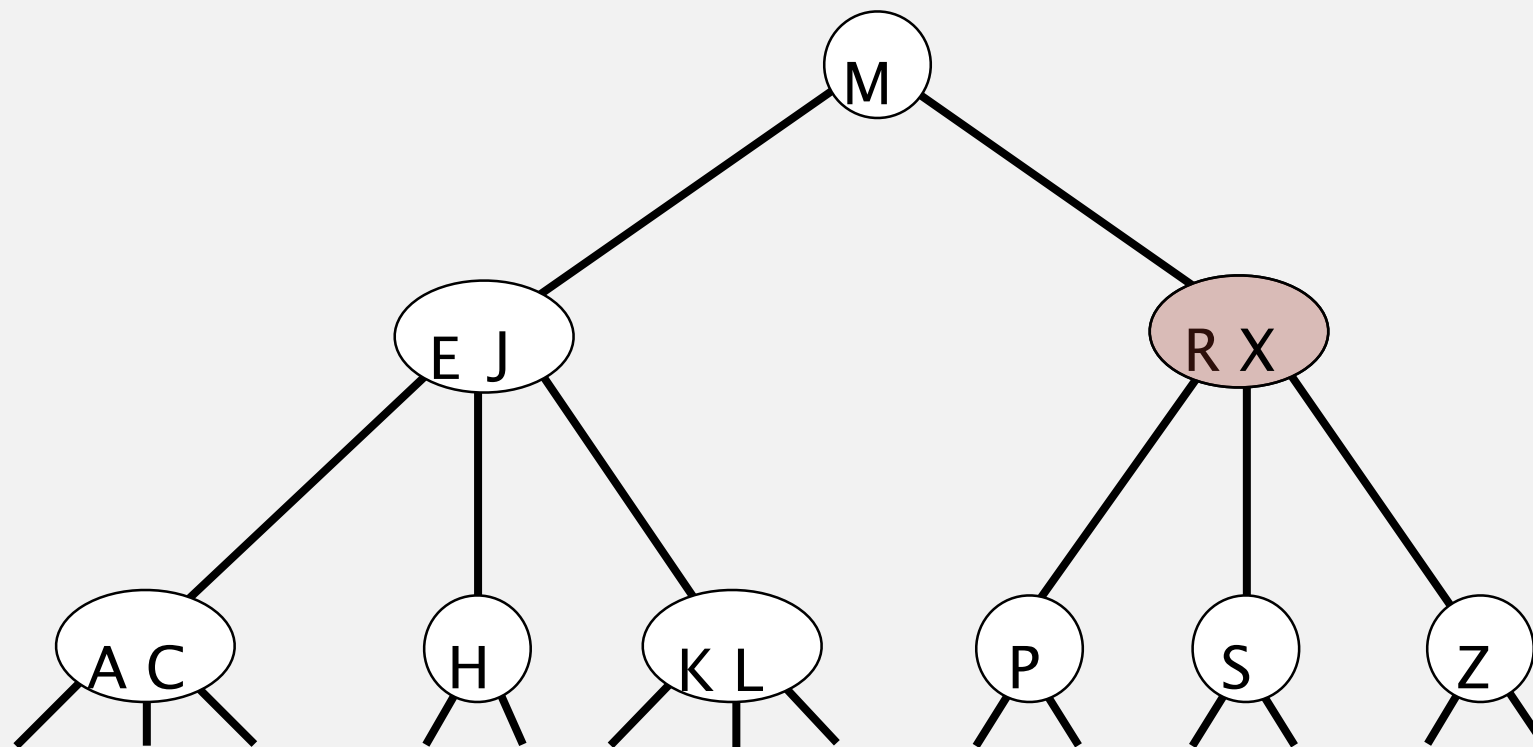- Move middle key in 4-node into parent.

insert Z

Z is greater than M

(go right)

M Z

E J

R

A C

H

K L

P

S X

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
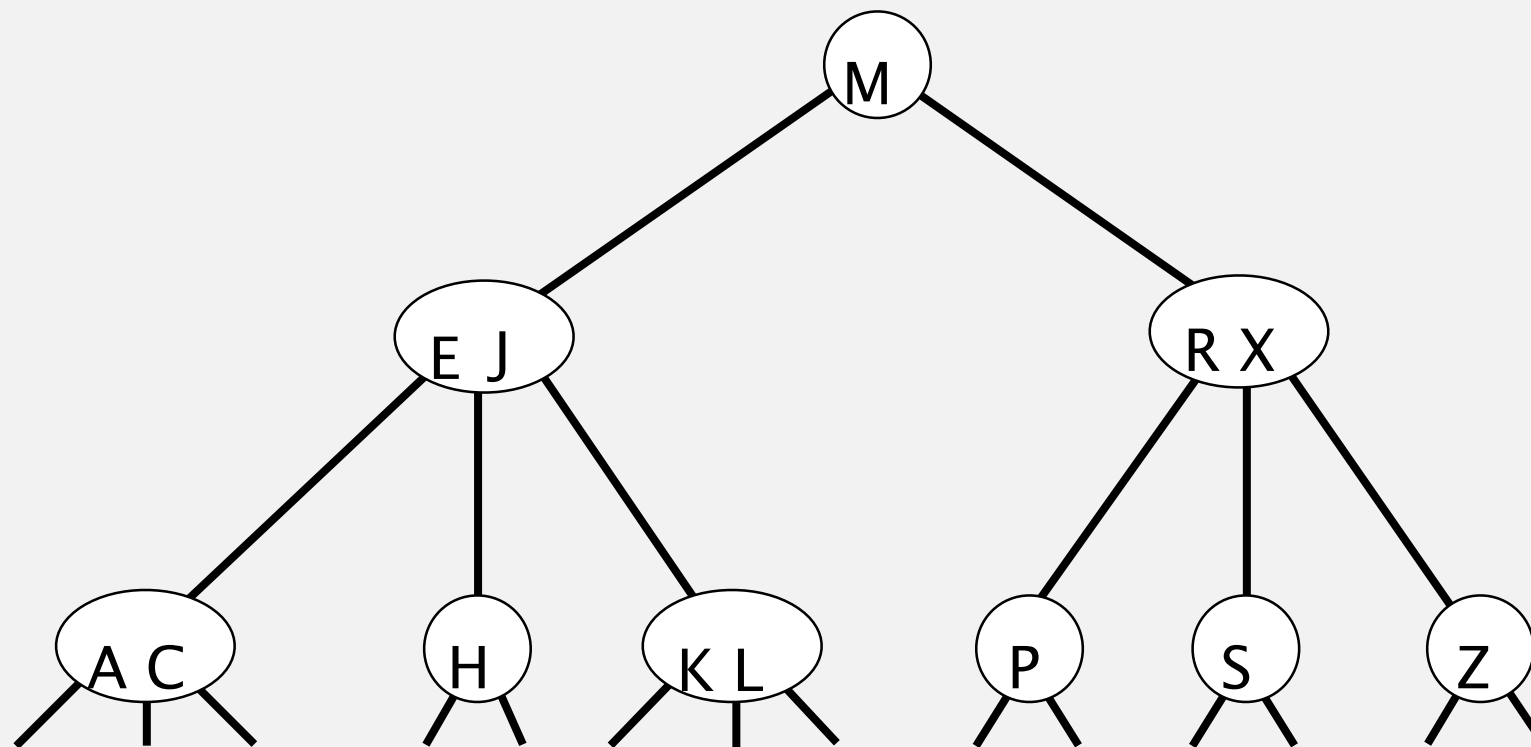- Move middle key in 4-node into parent.

insert Z

Z is greater than R
(go right)
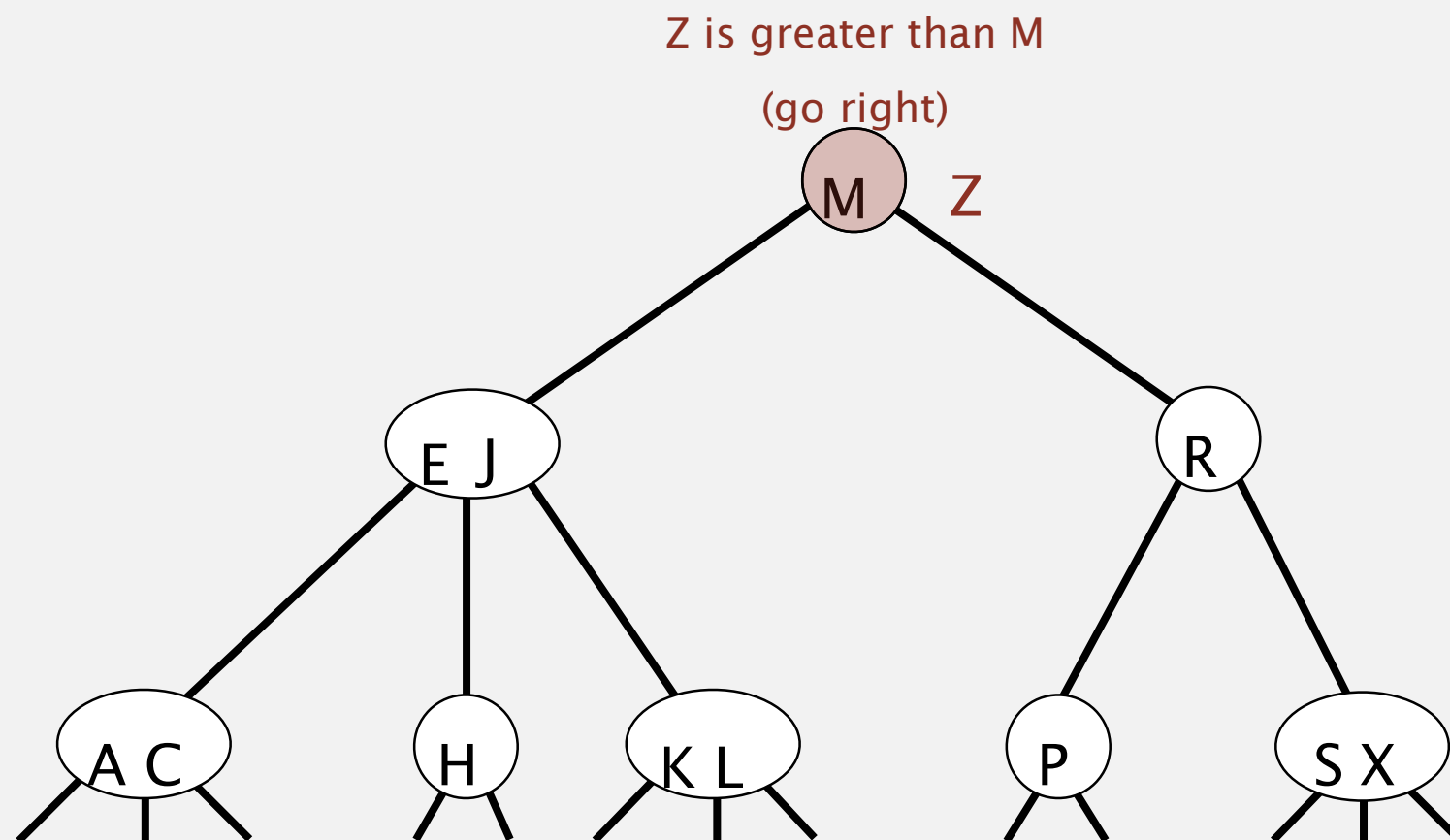
Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.

insert Z

search ends here

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
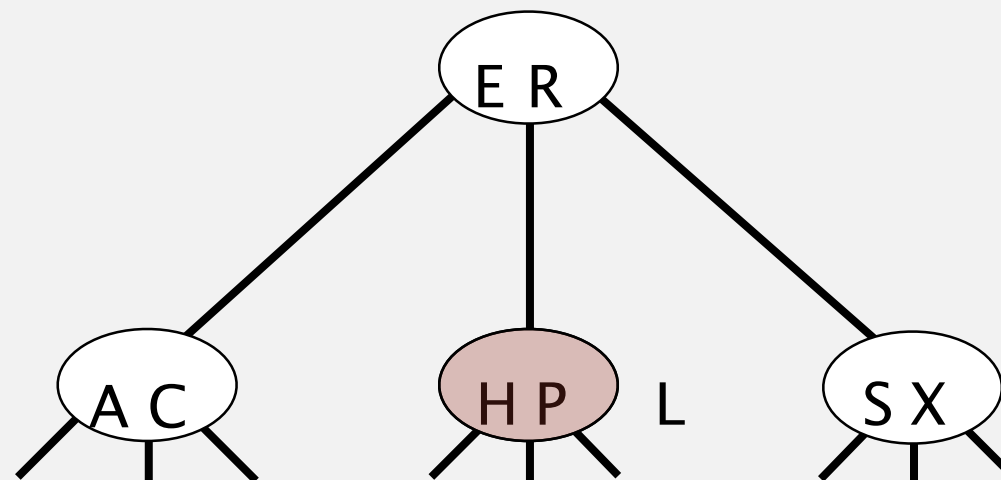- Move middle key in 4-node into parent.

insert Z

replace 3-node with

temporary 4-node containing Z

Insert into a 3-node at bottom.
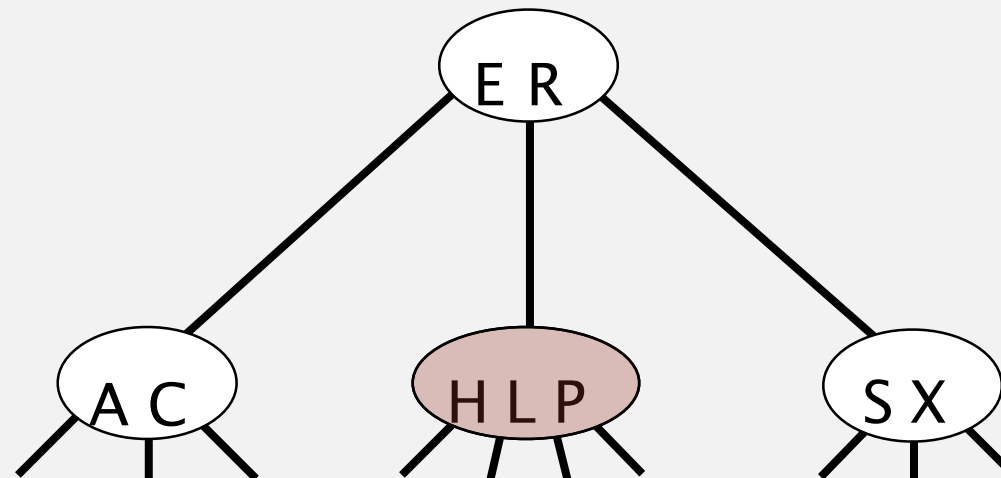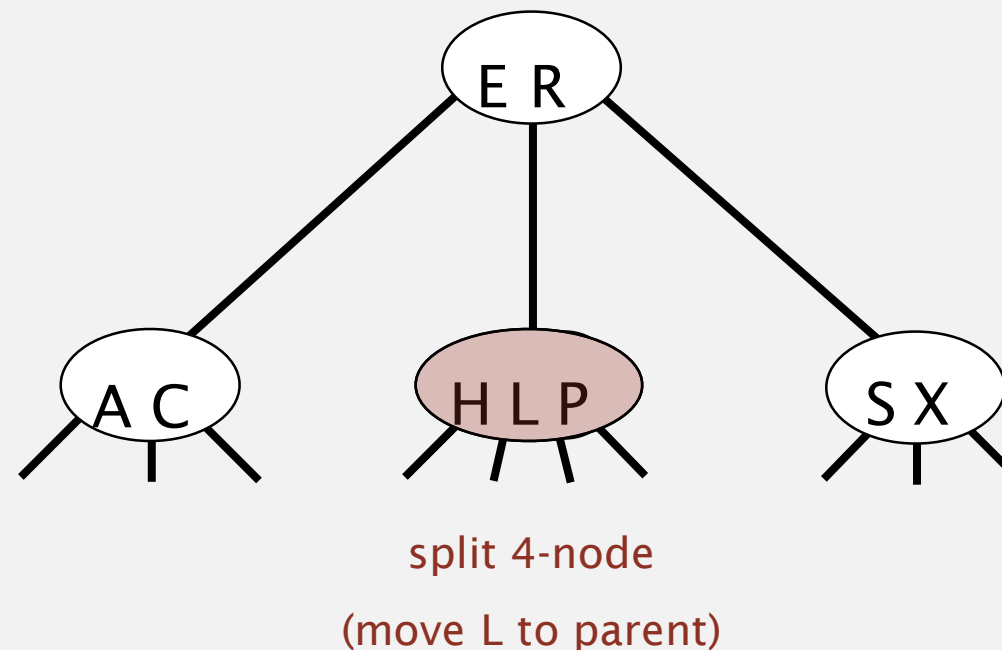
- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
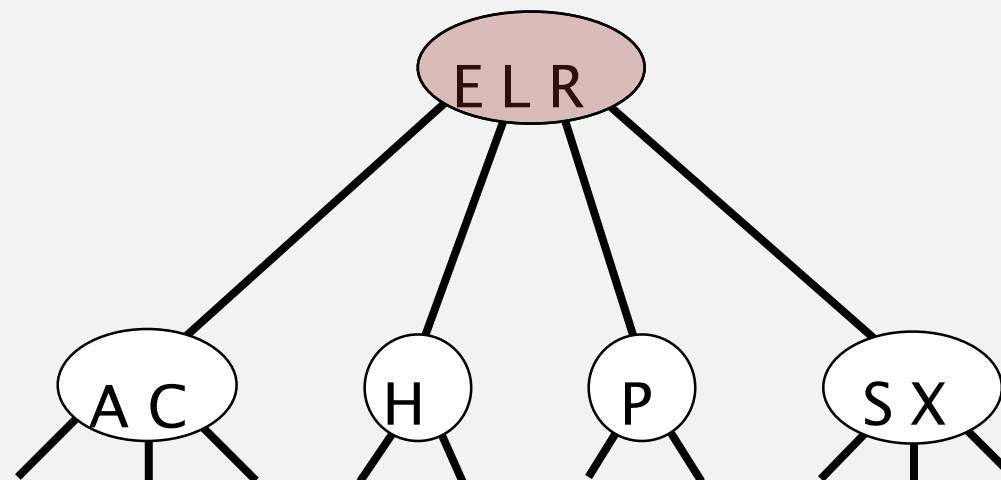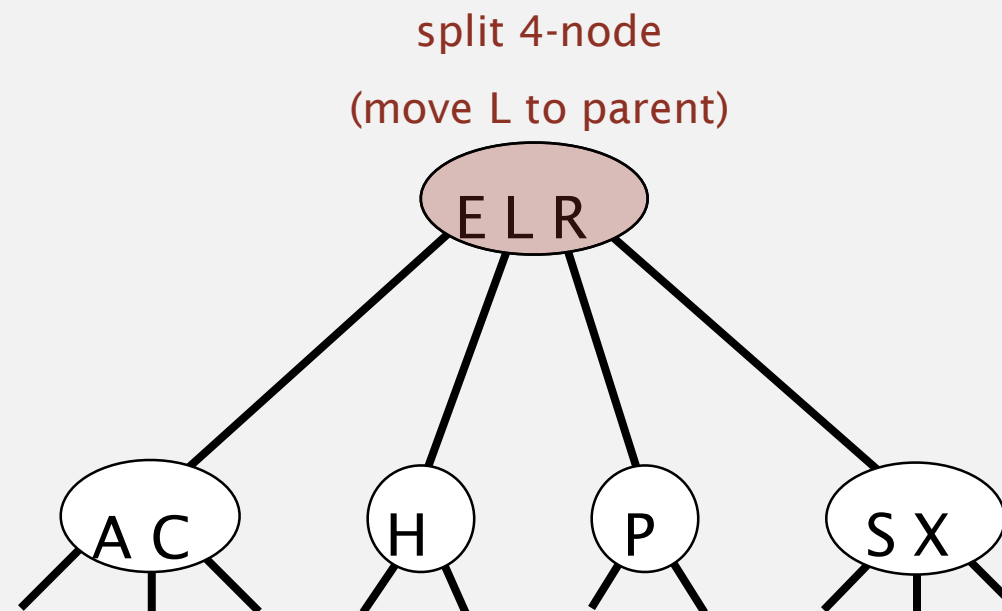
insert Z

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.

insert Z

split 4-node into two 2-nodes
(pass middle key to parent)

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
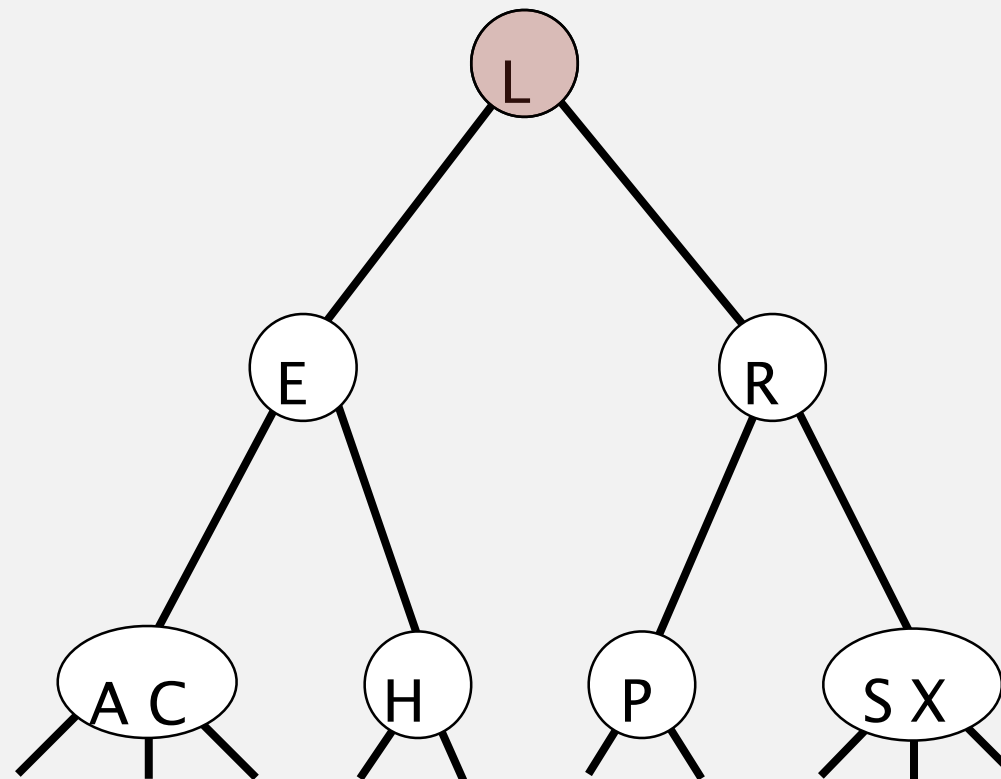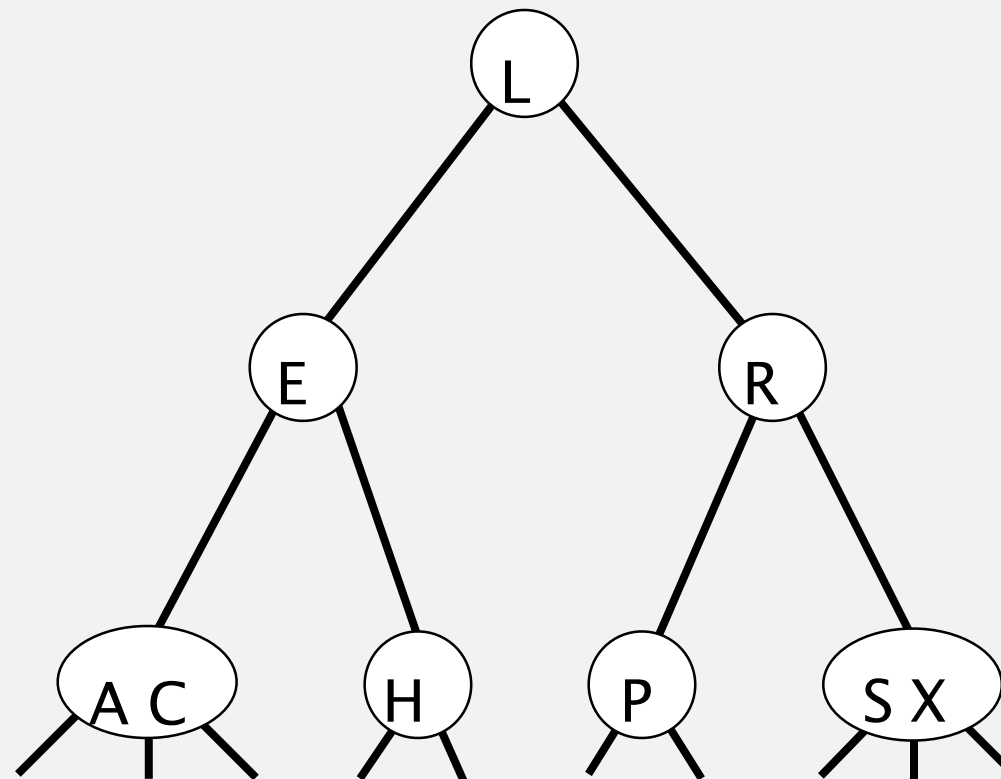
insert Z

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.

insert Z

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.
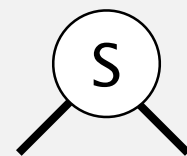
insert L

convert 3-node into 4-node

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.

insert L

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.

insert L

split 4-node

(move L to parent)

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.

insert L

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.

insert L

split 4-node

(move L to parent)

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.

height of tree increases by 1

insert L

Insert into a 3-node at bottom.

- Add new key to 3-node to create temporary 4-node.
- Move middle key in 4-node into parent.
- Repeat up the tree, as necessary.
- If you reach the root and it's a 4-node, split it into three 2-nodes.

insert L

3.3 2-3 Tree Demo

‣ *search*

‣ *insertion*

‣ **construction**

Algorithms

Robert Sedgewick | Kevin Wayne

insert S

2-3 tree

insert E

E S
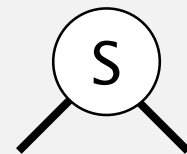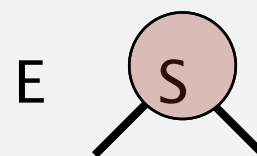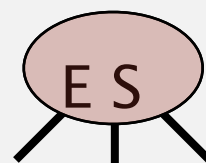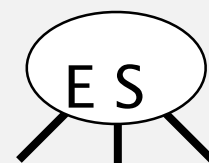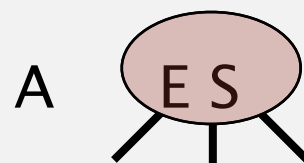
convert 2-node into 3-node

insert E

2-3 tree

insert A

convert 3-node into 4-node
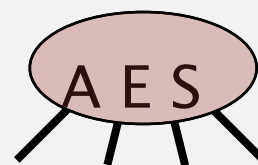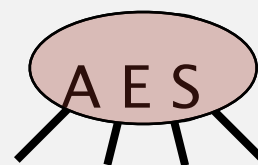
insert A

insert A

split 4-node
(move E to parent)
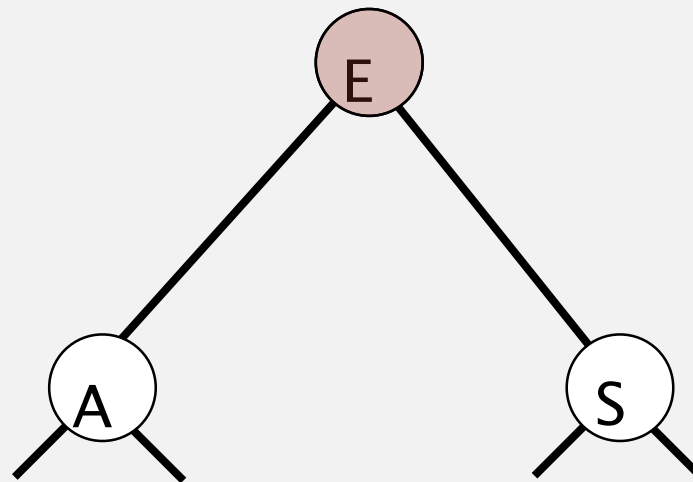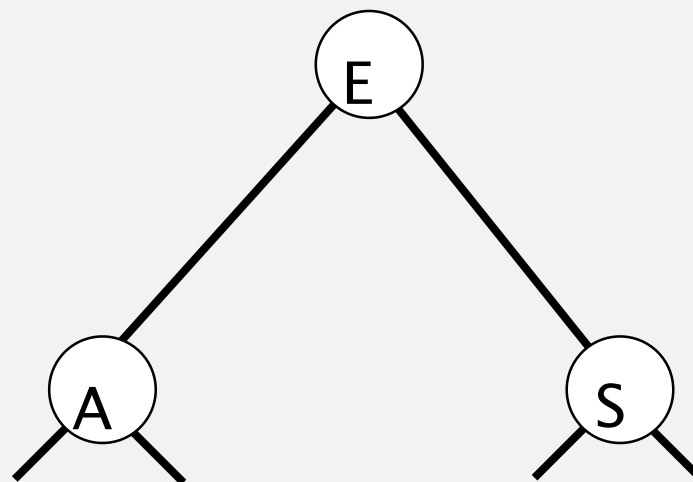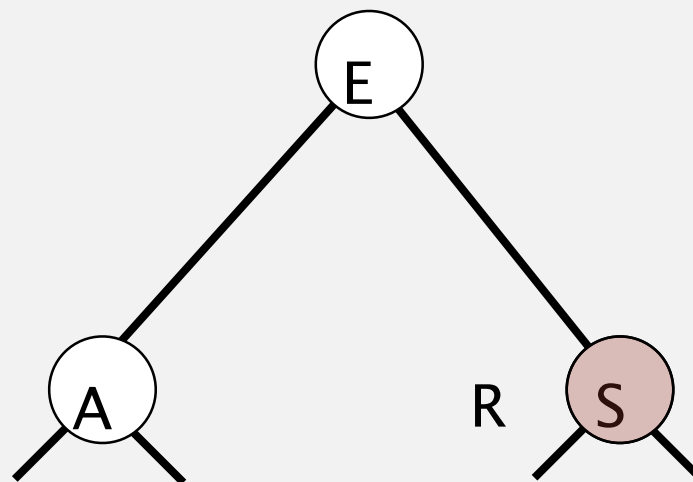
insert A

2-3 tree

insert R

convert 2-node into 3-node
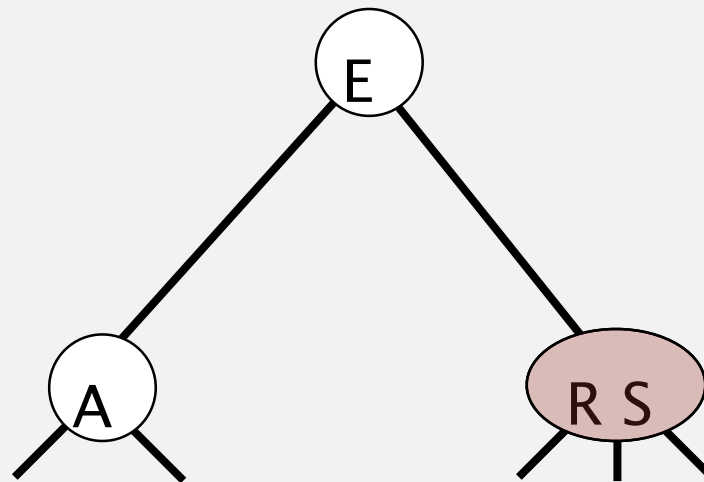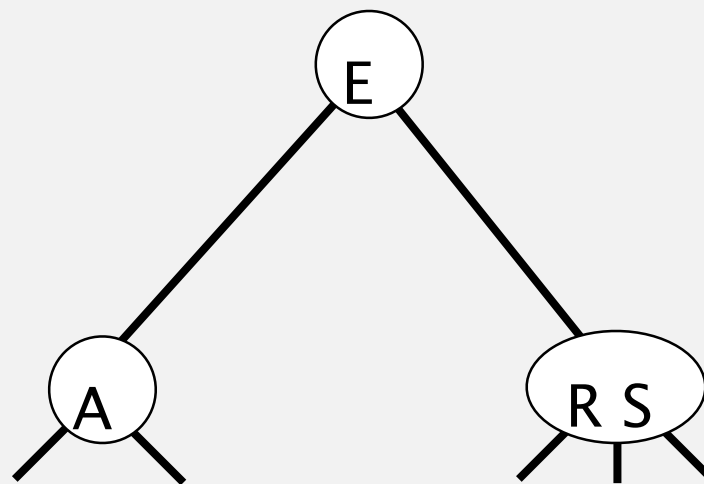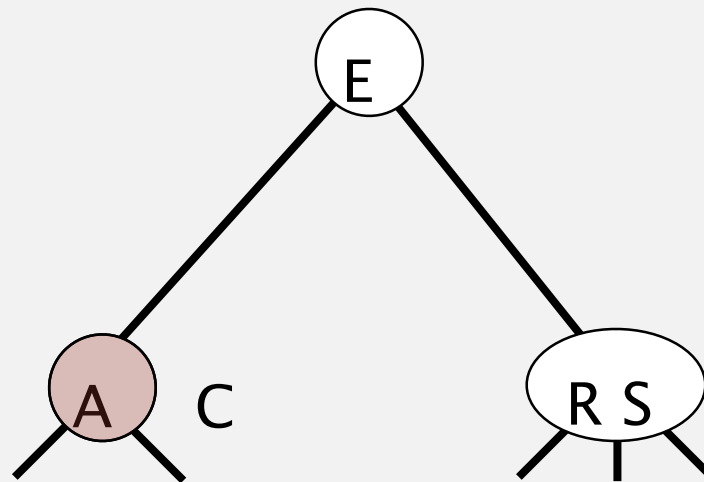
insert R

2-3 tree

insert C

convert 2-node into 3-node
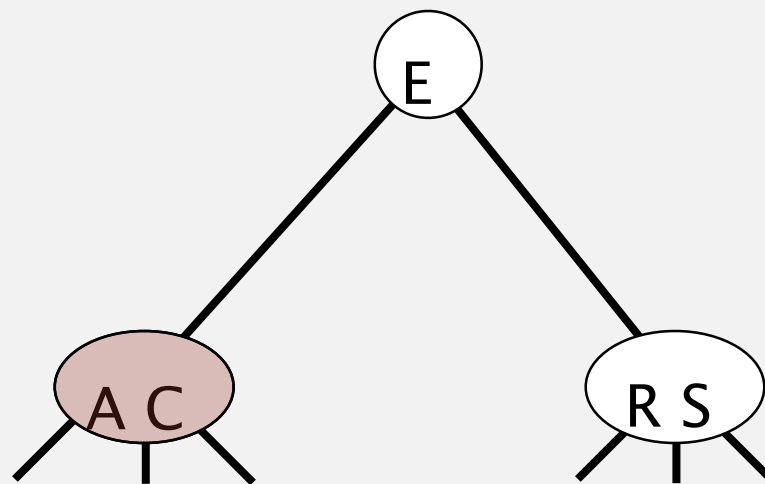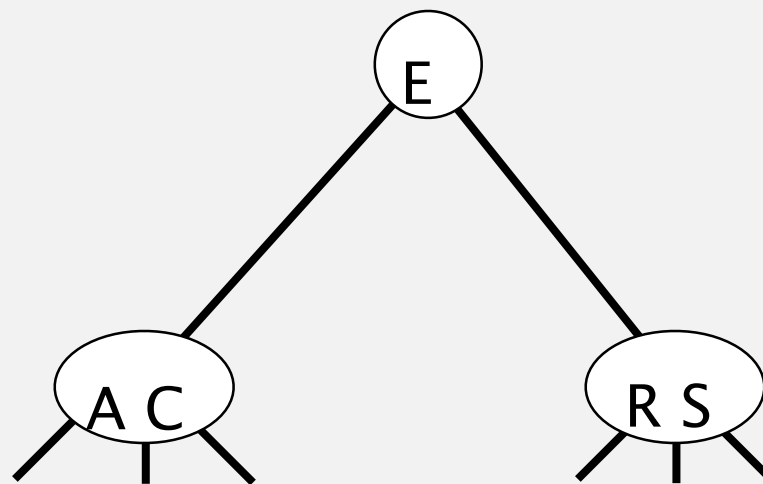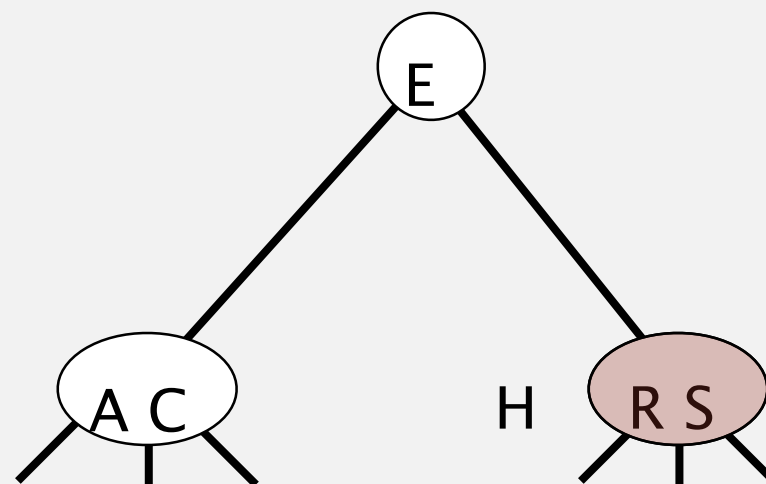
insert C

2-3 tree

insert H

convert 3-node into 4-node
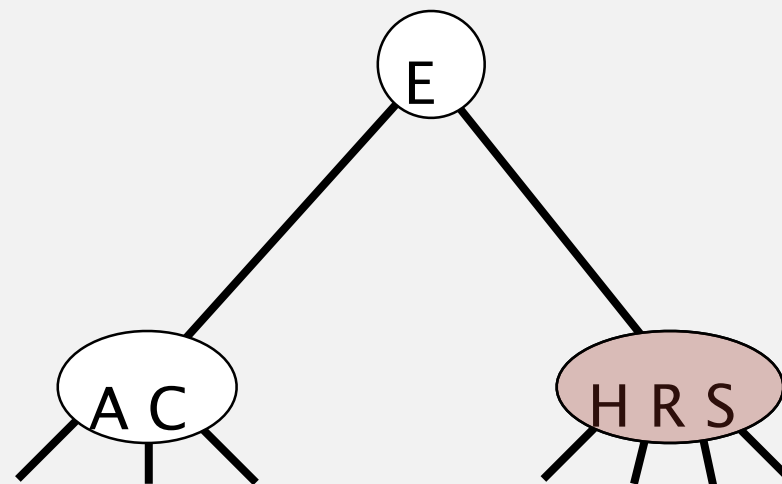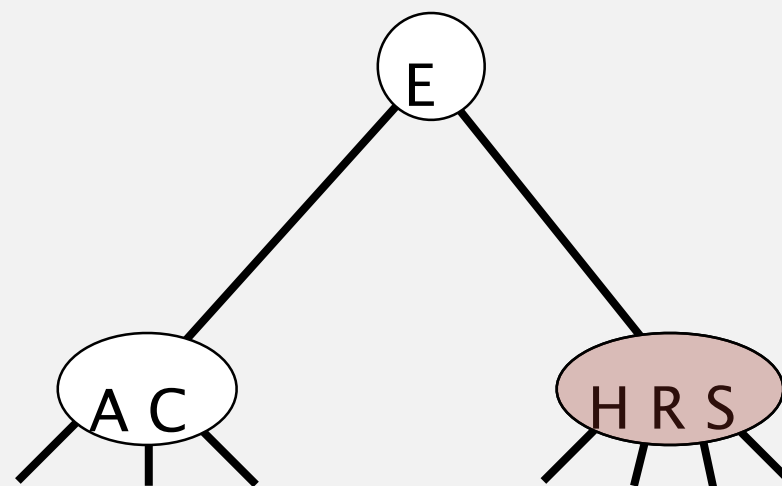
insert H

insert H

split 4-node
(move R to parent)
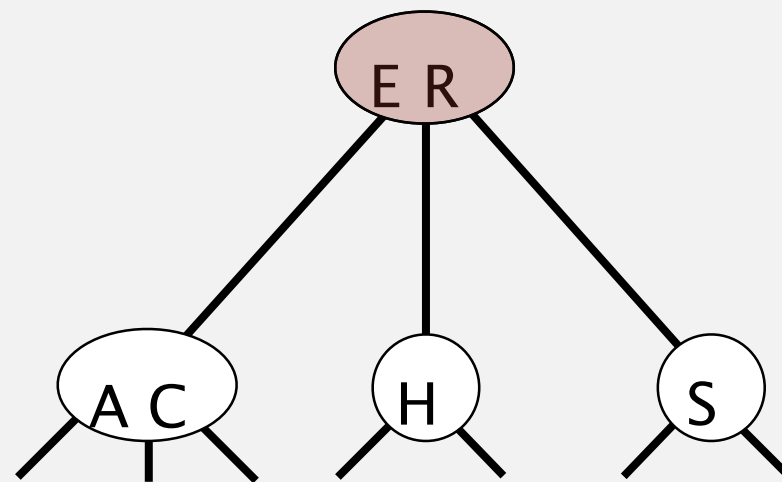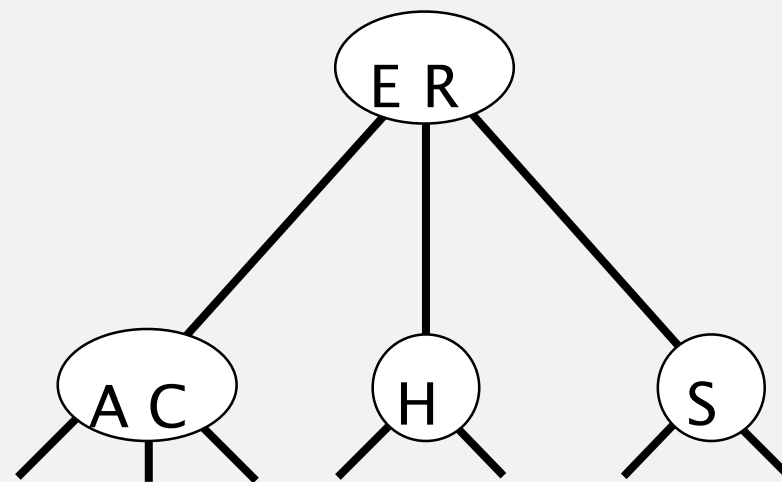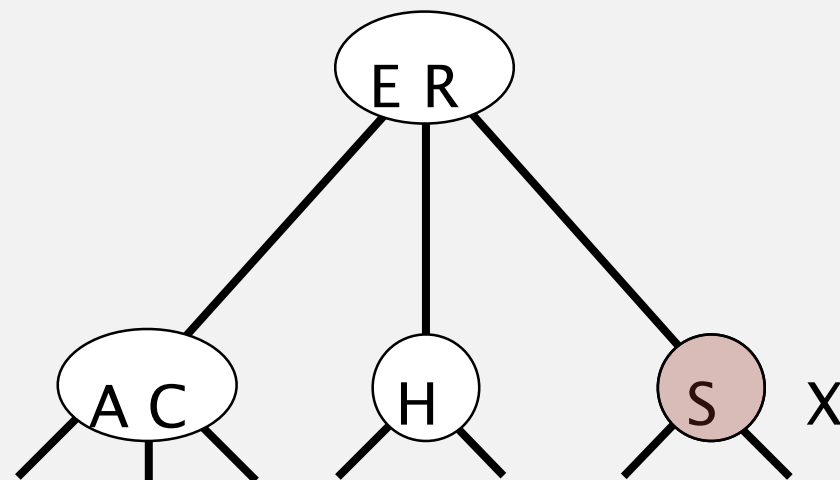
insert H

2-3 tree

insert X

convert 2-node into 3-node
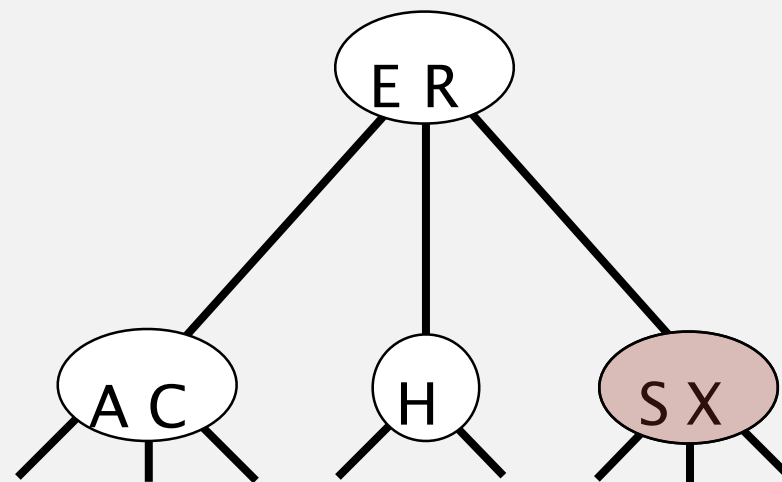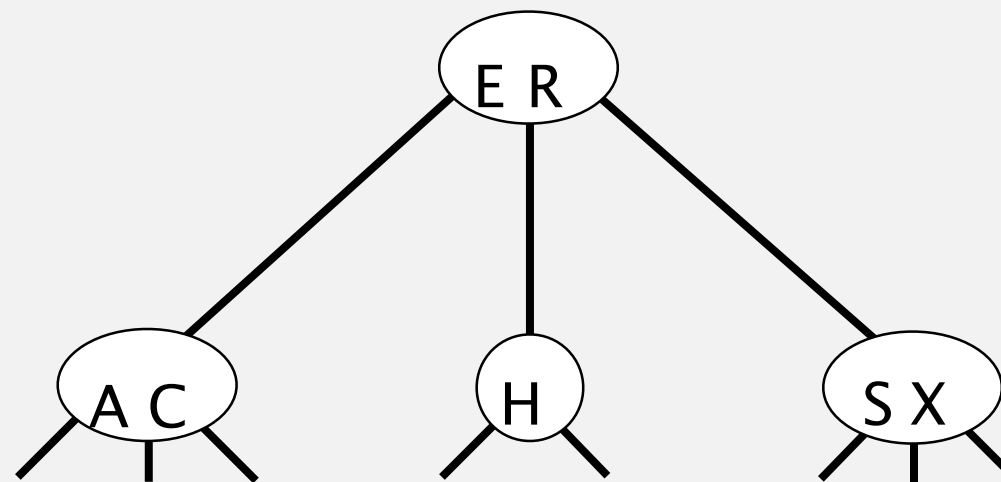
insert X

2-3 tree

insert P

convert 2-node into 3-node

insert P

2-3 tree

insert L

convert 3-node into 4-node

insert L

insert L

E R

A C   H L P   S X

split 4-node

(move L to parent)

insert L

insert L

split 4-node

(move L to parent)

E L R

A C   H   P   S X

insert L

2-3 tree