Algorithms

Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

3.3 2-3 Tree Demo

- *search*
- *insertion*
- *construction*

Algorithms

Robert Sedgewick | Kevin Wayne

http://algs4.cs.princeton.edu

3.3 2-3 Tree Demo

- *search*
- *insertion*
- *construction*

2-3 tree demo: search

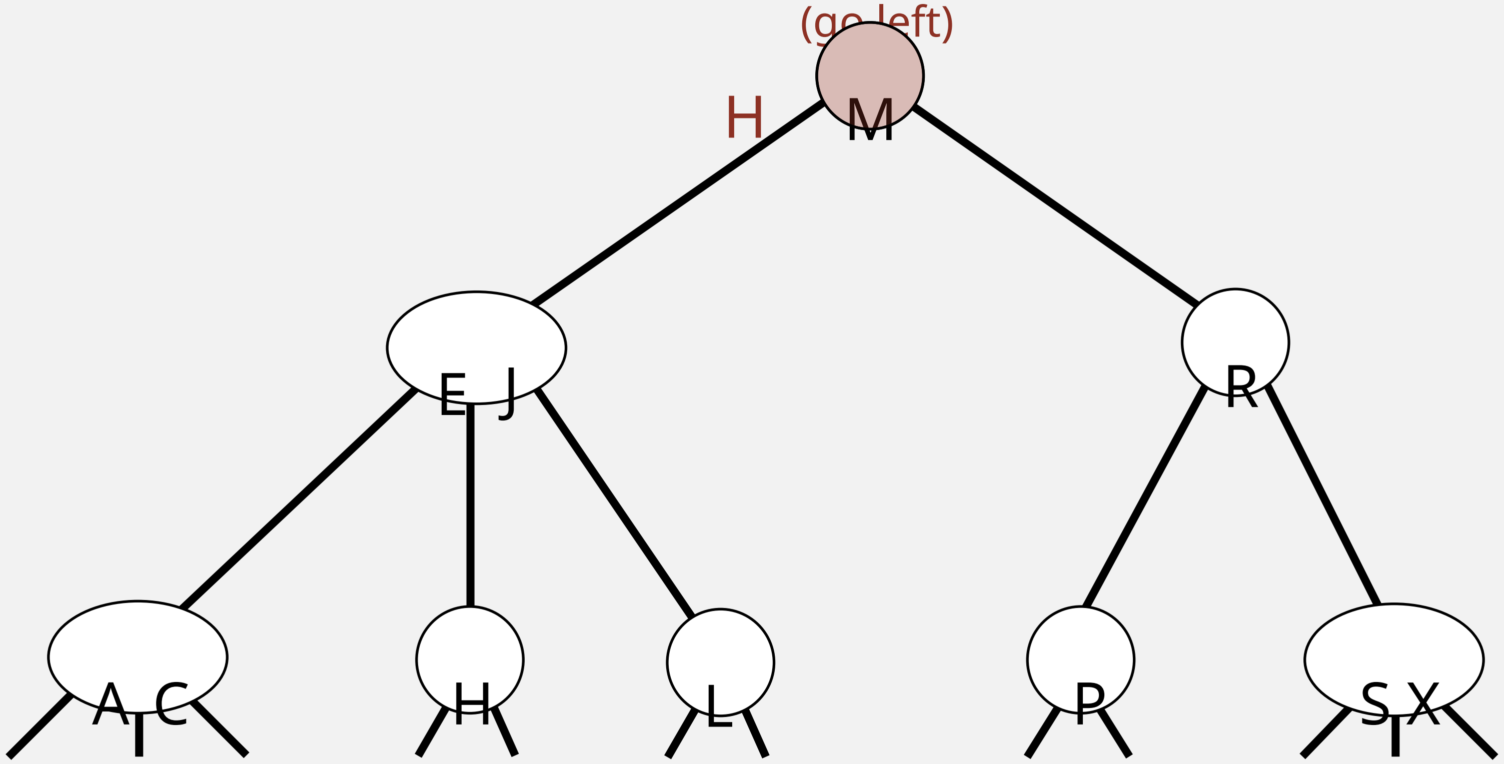
Search.

- 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)

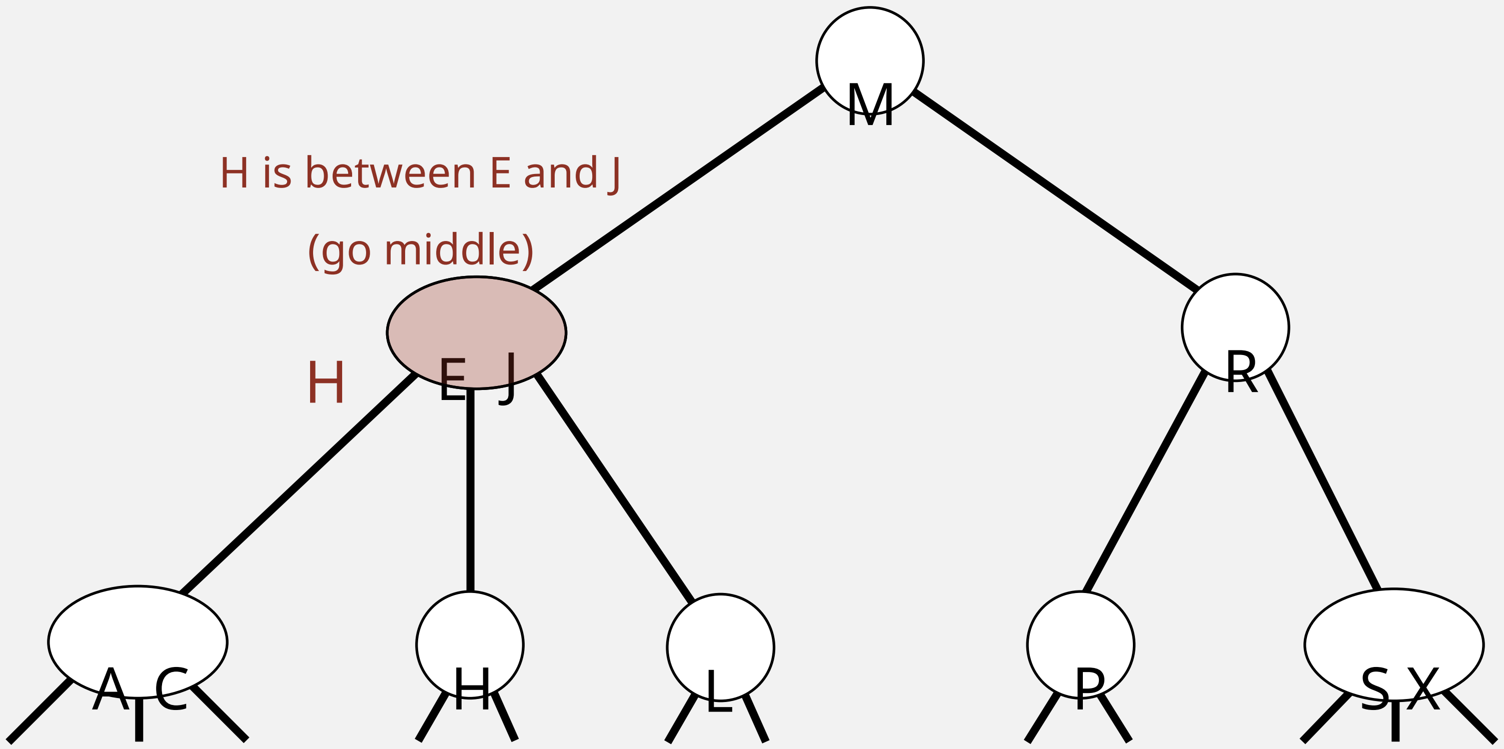


2-3 tree demo: search

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

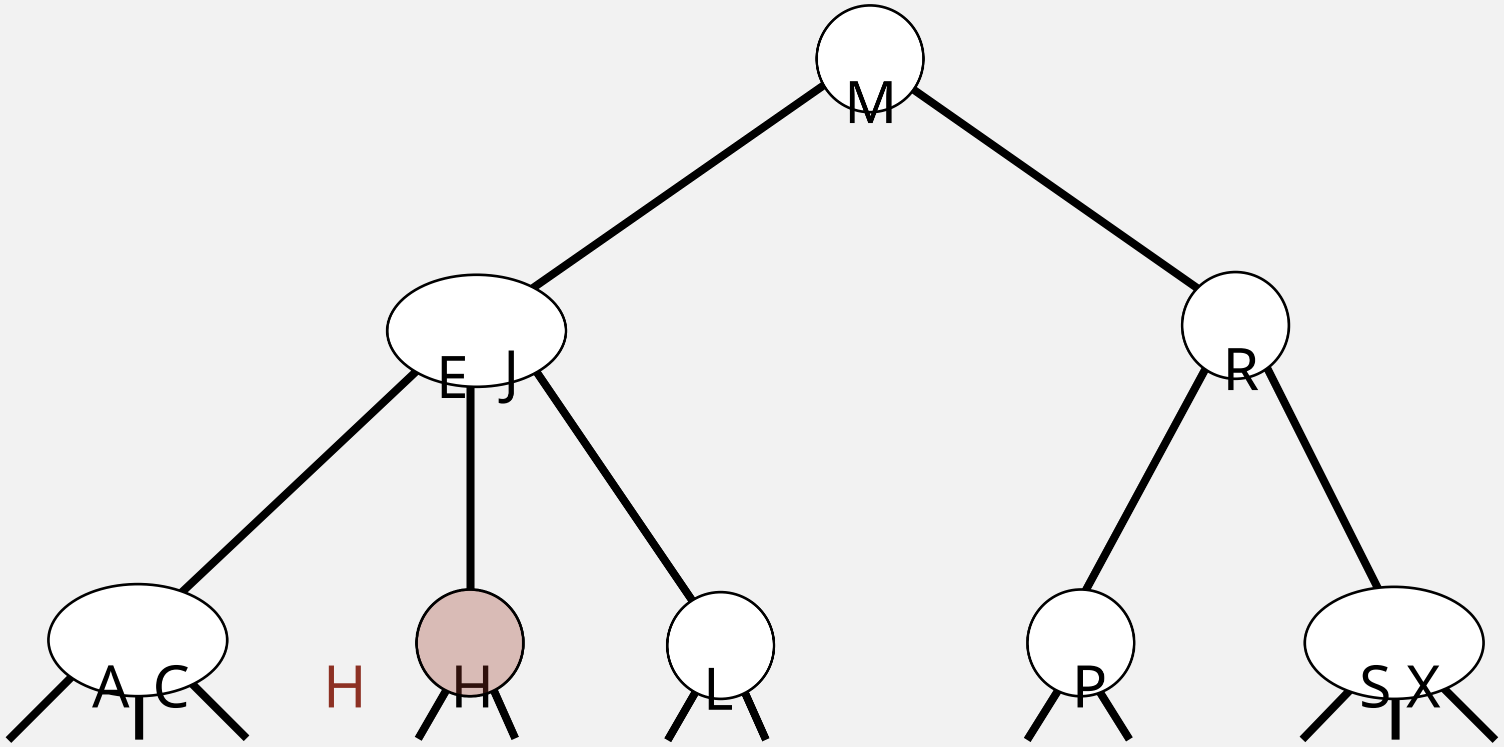
(go middle)

2-3 tree demo: search

Search.

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

search for H



found H

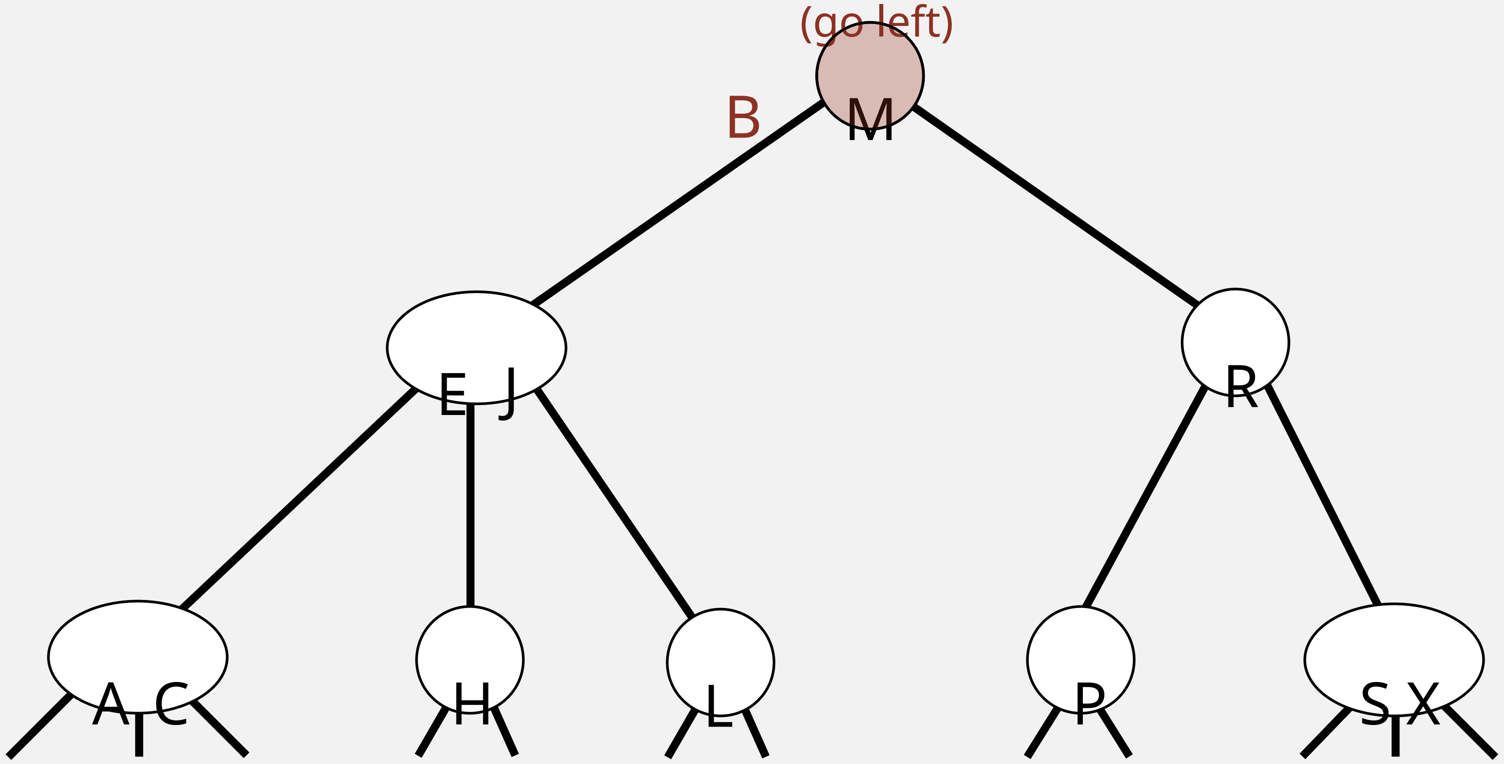
(search hit)

2-3 tree demo: search

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

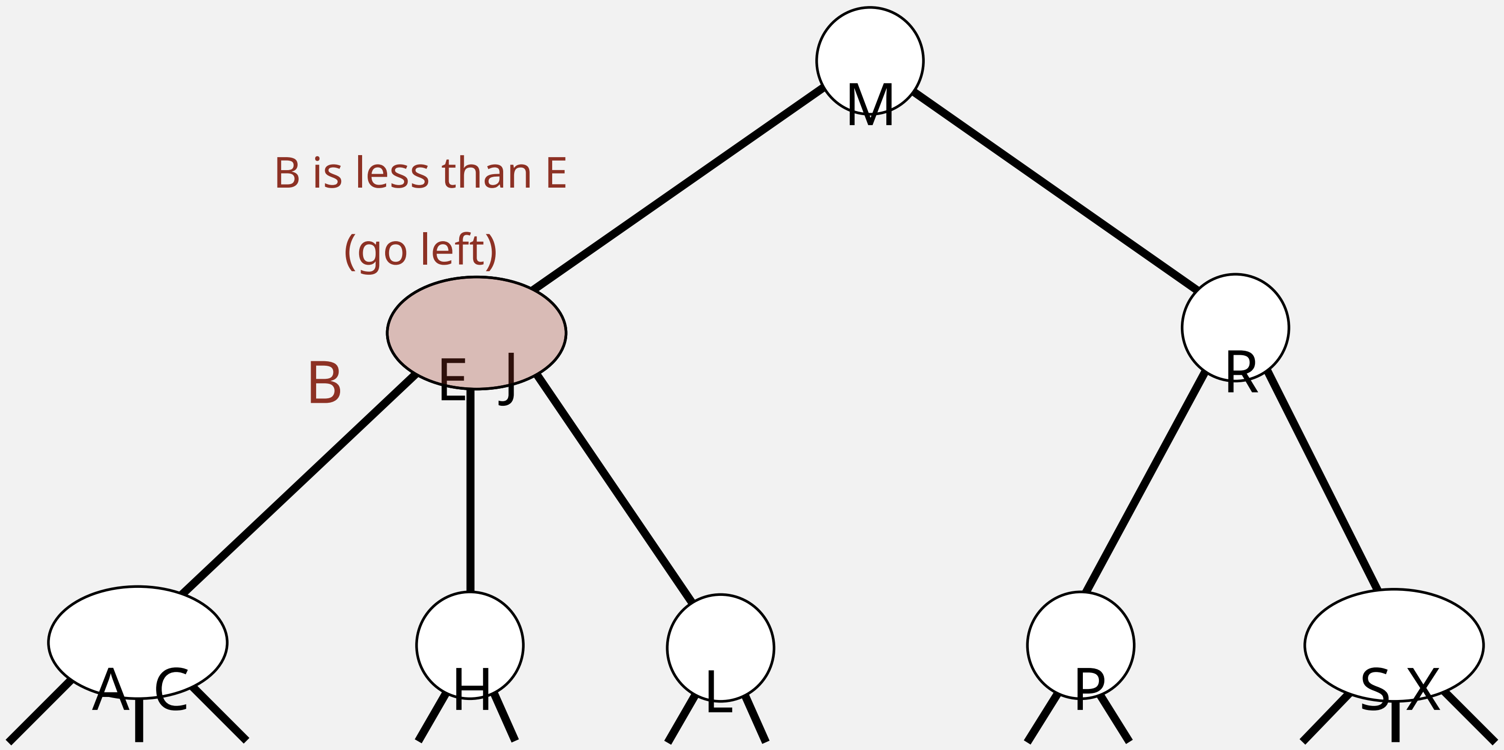
(go left)

2-3 tree demo: search

Search.

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

search for B

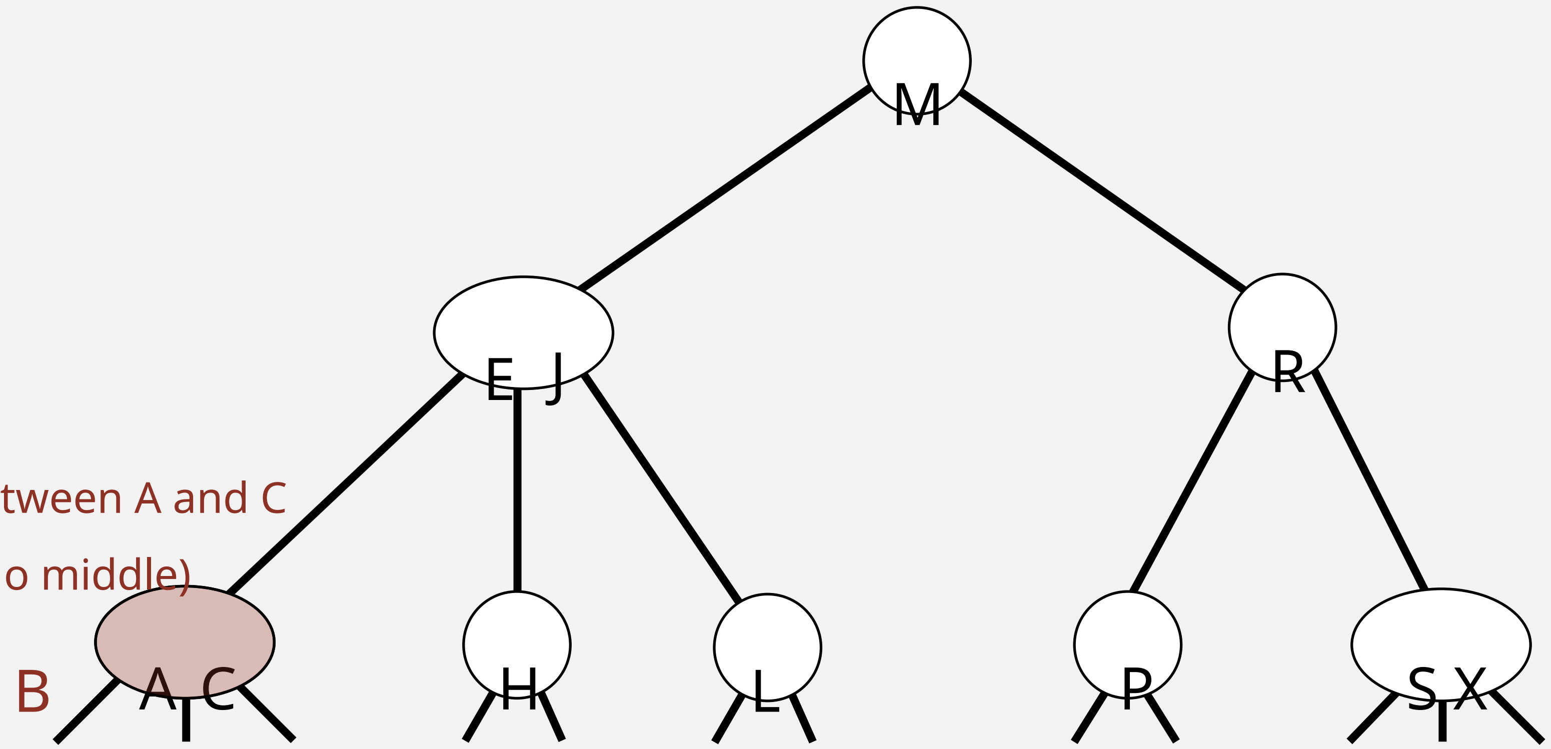


2-3 tree demo: search

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

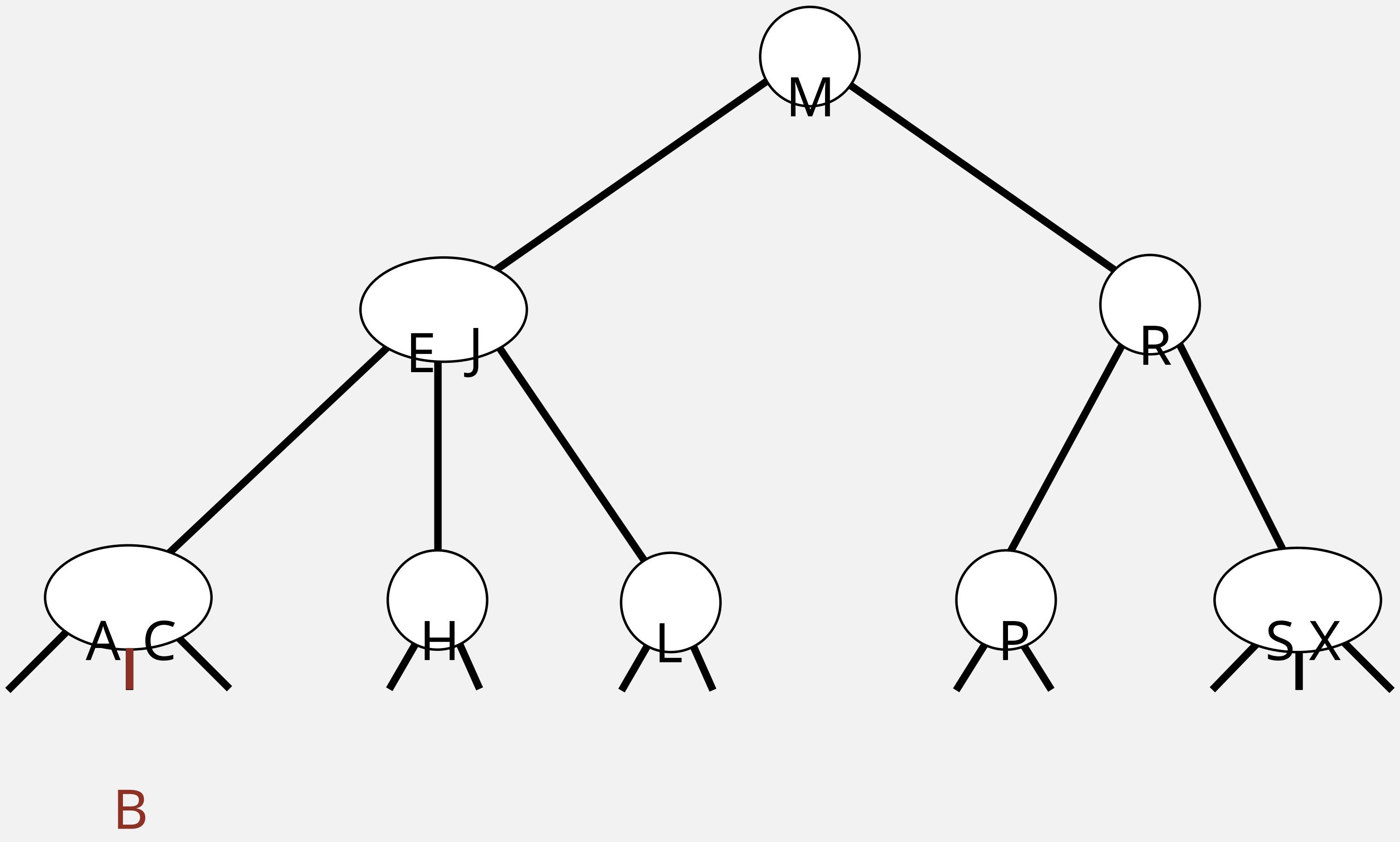
(go middle)

2-3 tree demo: search

Search.

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

search for B



link is null

(search miss)

http://algs4.cs.princeton.edu

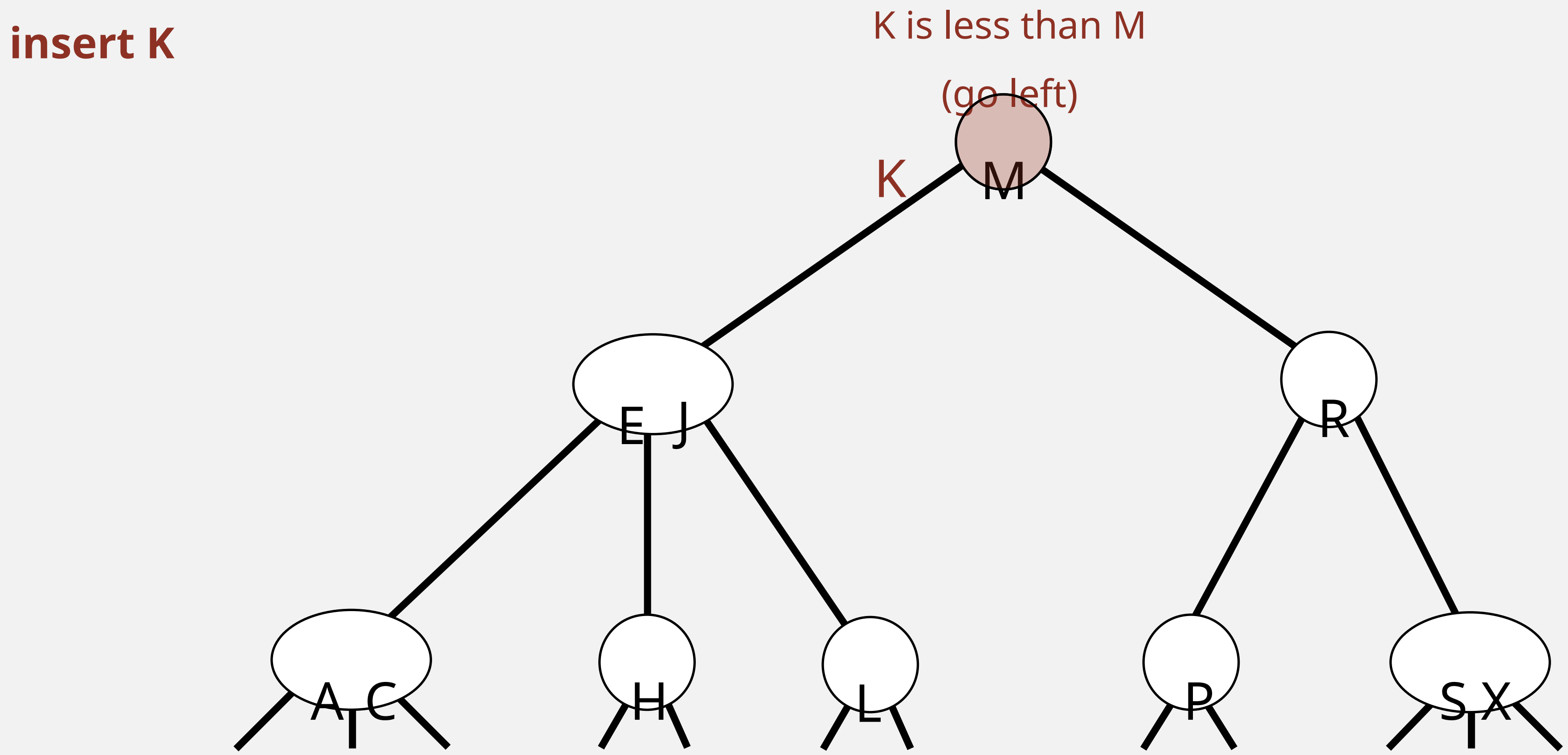
3.3 2-3 Tree Demo

- *search*
- *insertion*
- *construction*

2-3 tree demo: insertion

Insert into a 2-node at bottom.

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

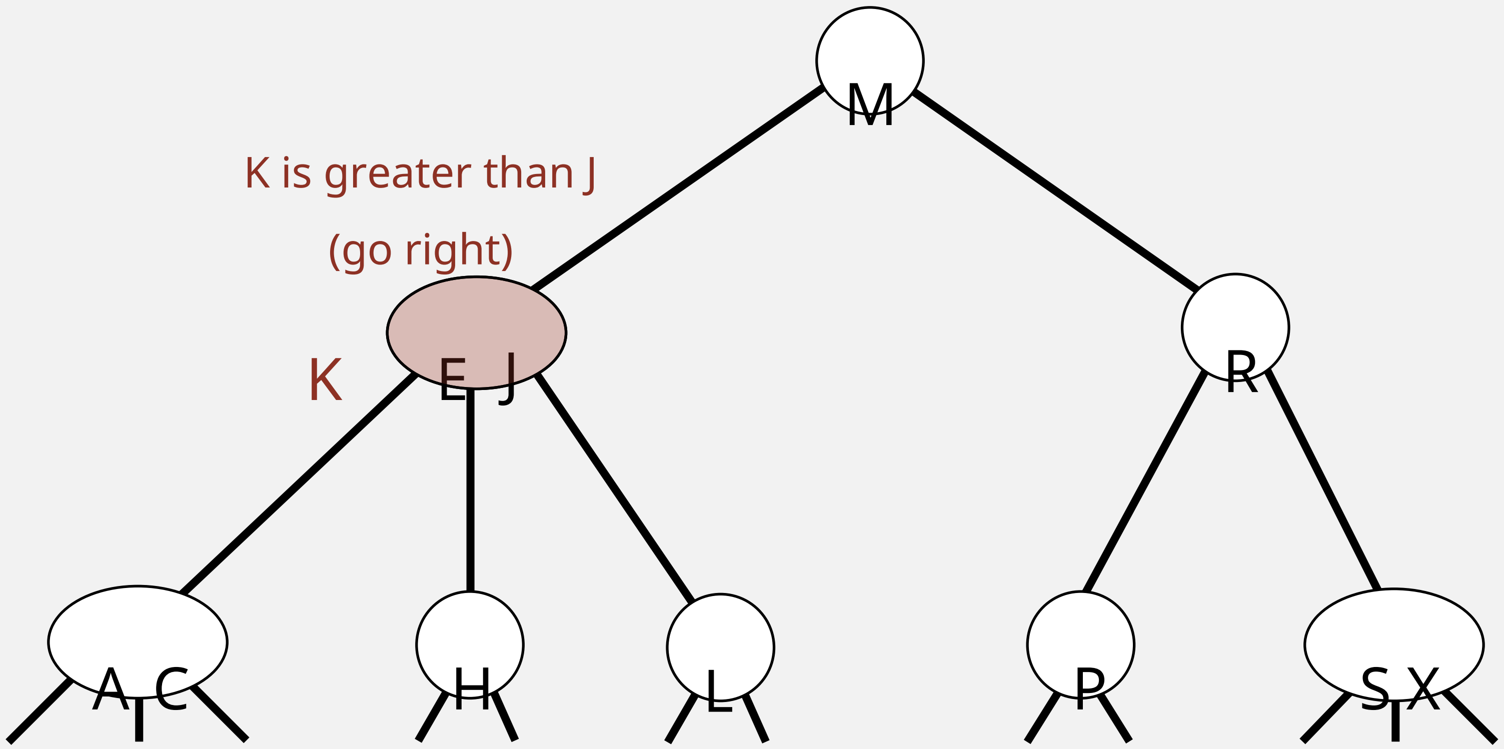


2-3 tree demo: insertion

Insert into a 2-node at bottom.

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

insert K

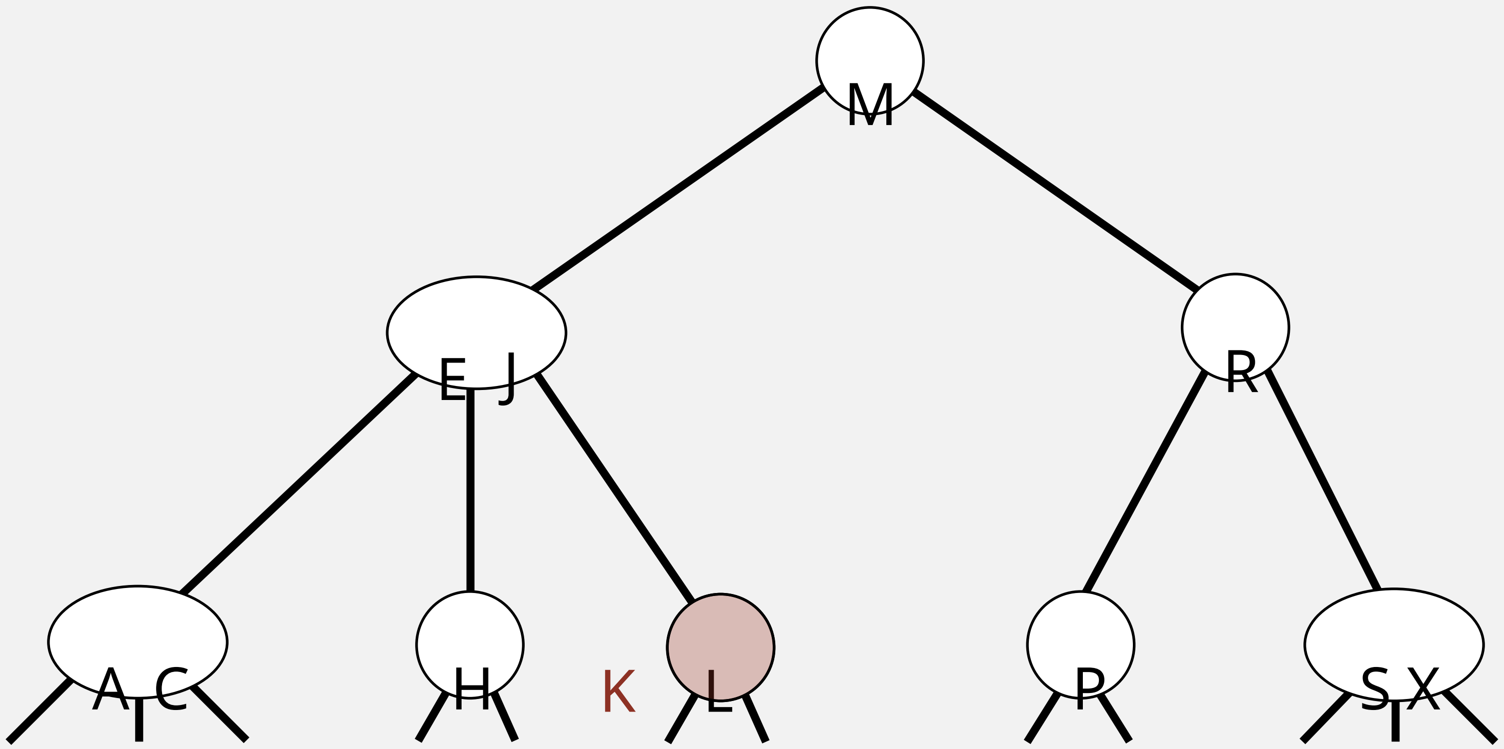


2-3 tree demo: insertion

Insert into a 2-node at bottom.

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

insert K



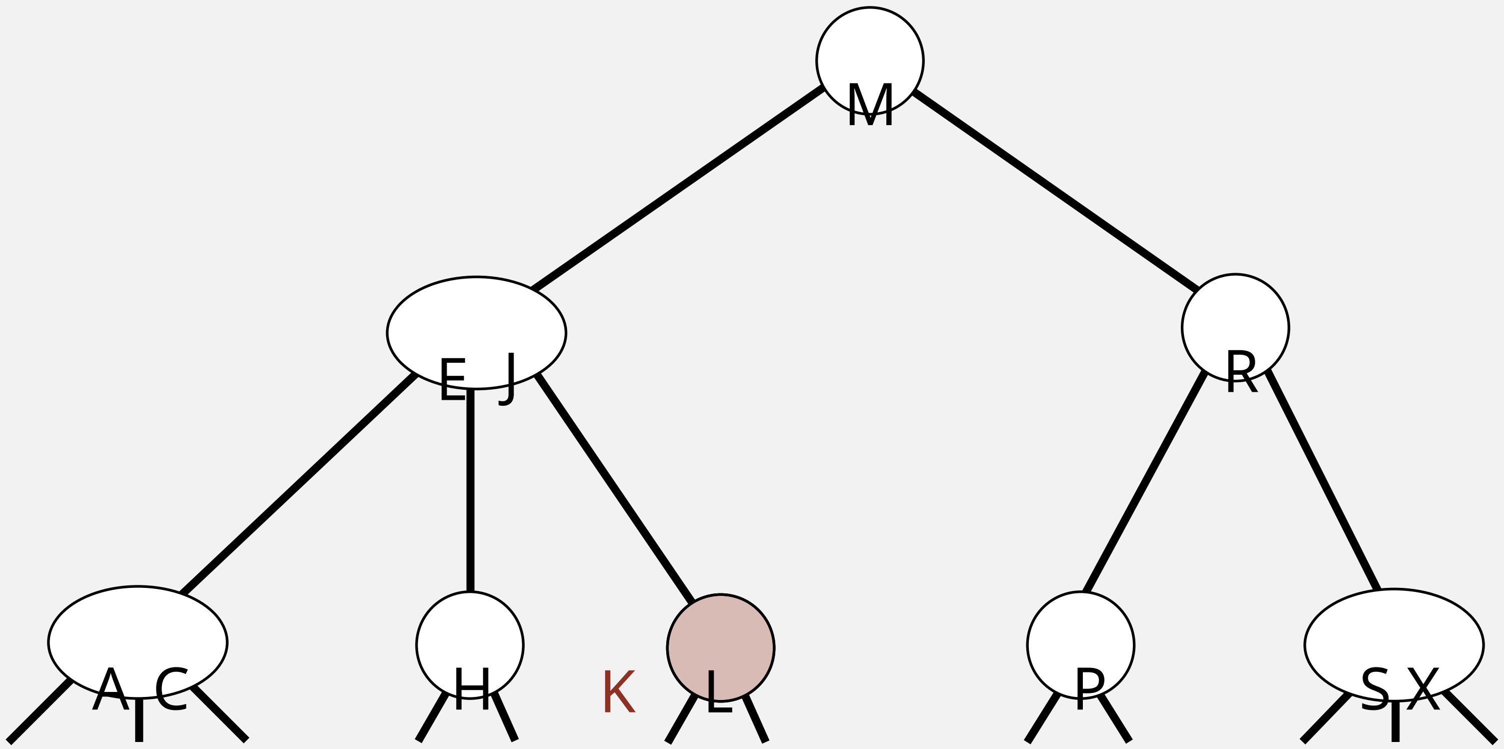
search ends here

2-3 tree demo: insertion

Insert into a 2-node at bottom.

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

insert K



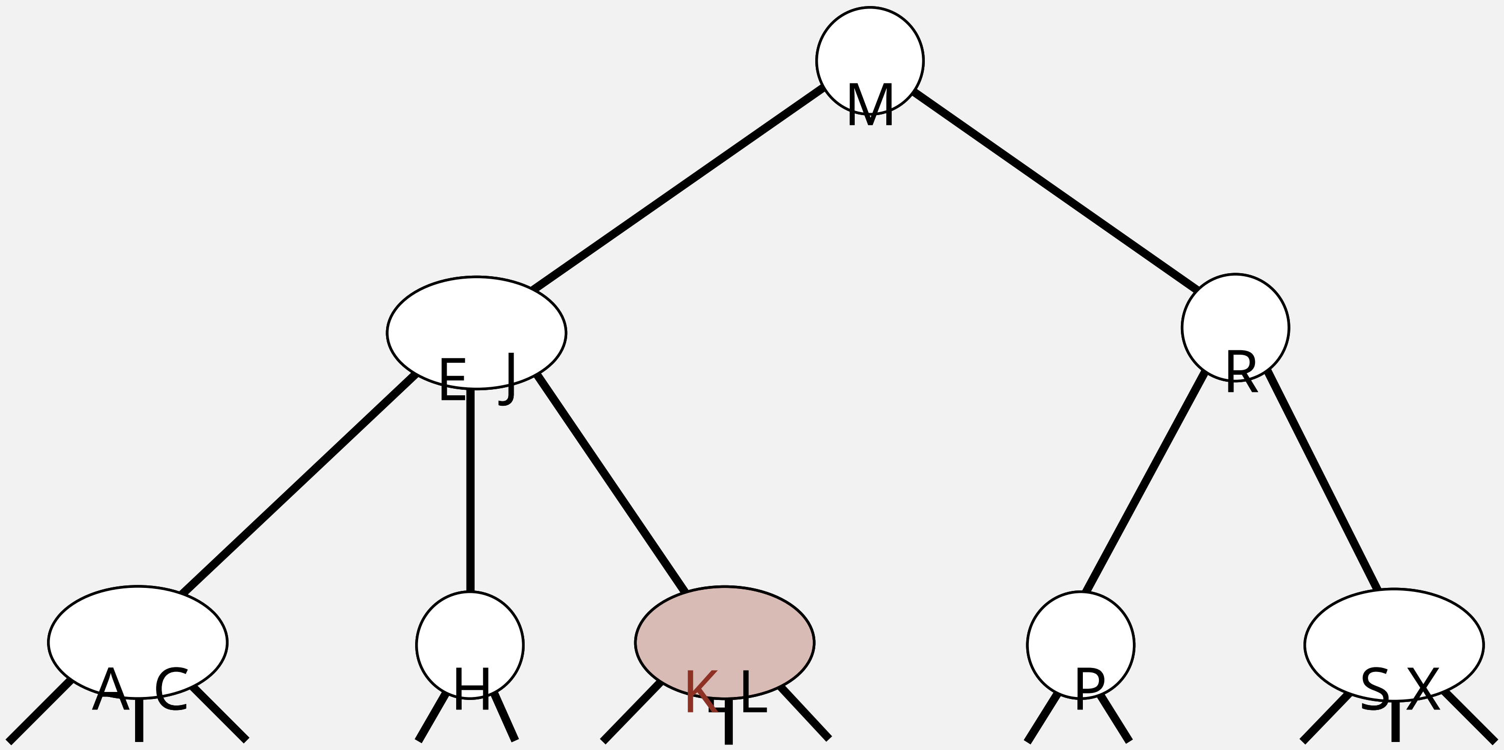
replace 2-node with
3-node containing K

2-3 tree demo: insertion

Insert into a 2-node at bottom.

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

insert K

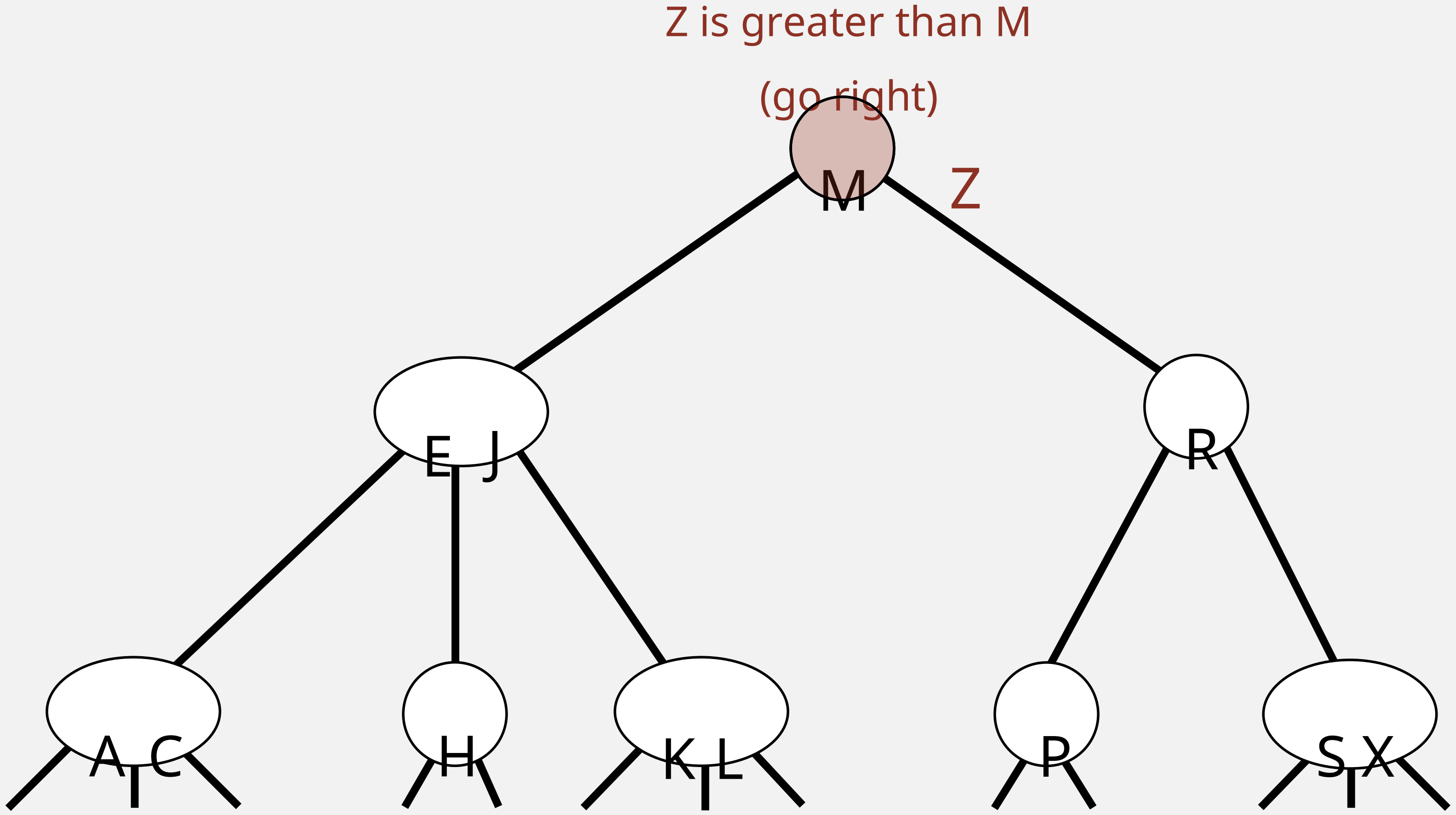


2-3 tree demo: insertion

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

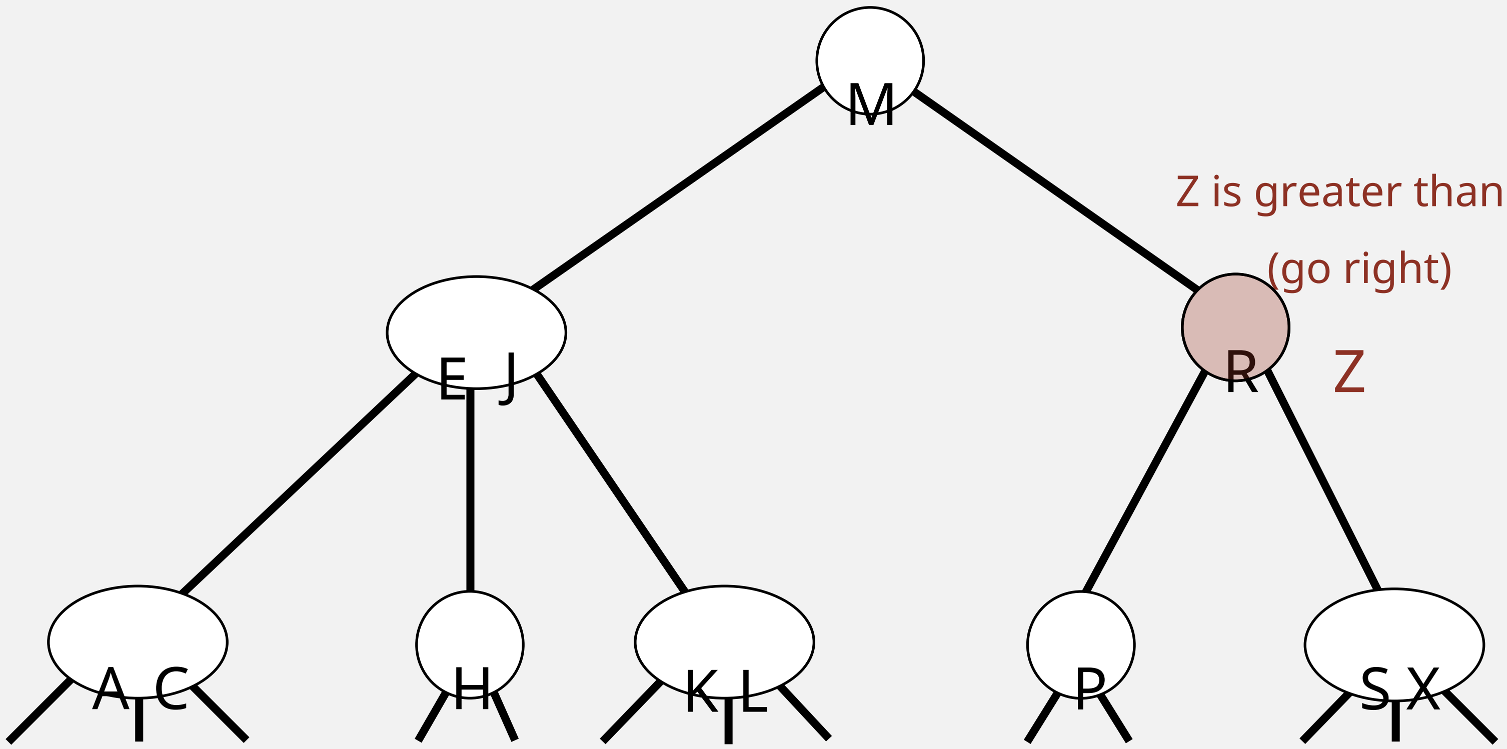


2-3 tree demo: insertion

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

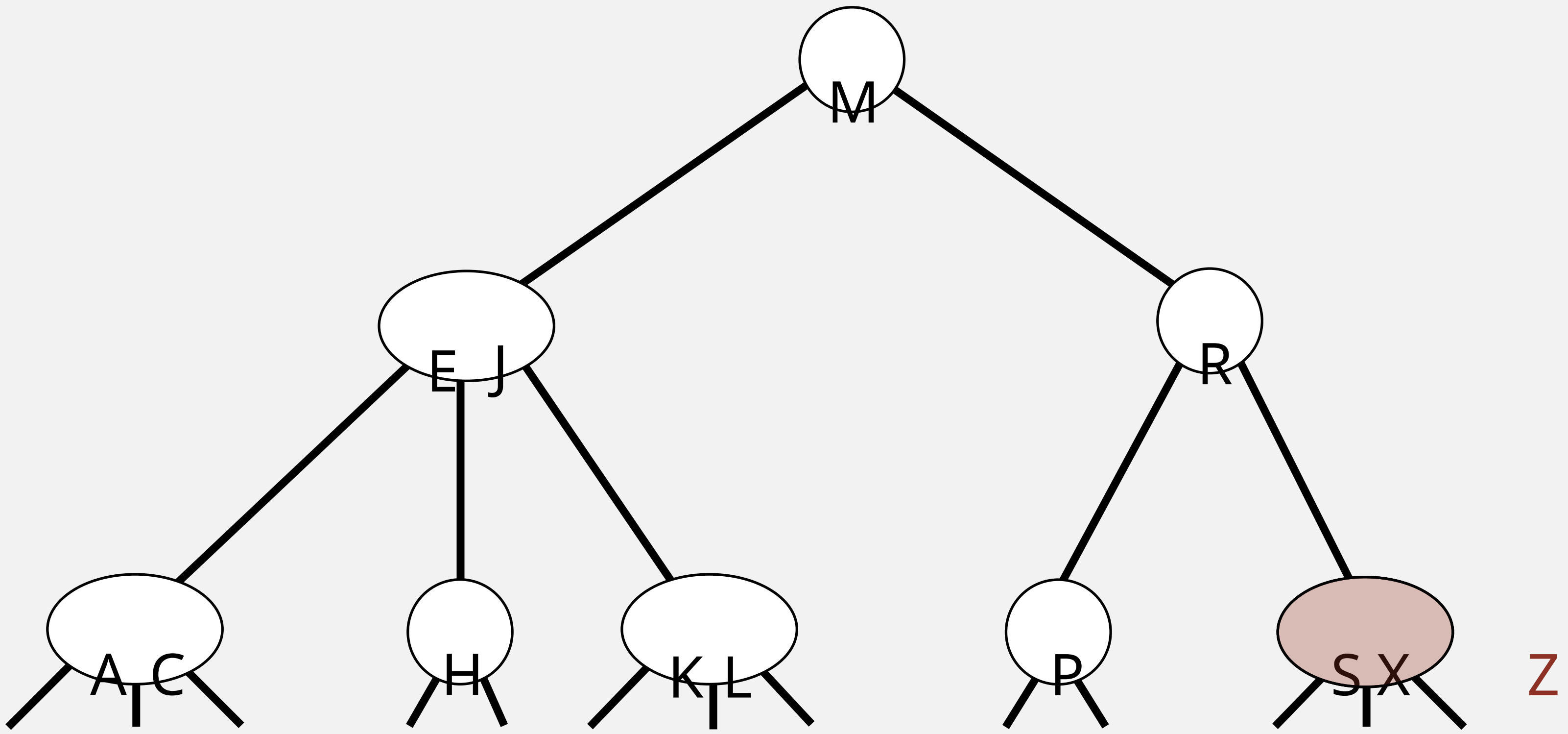


2-3 tree demo: insertion

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



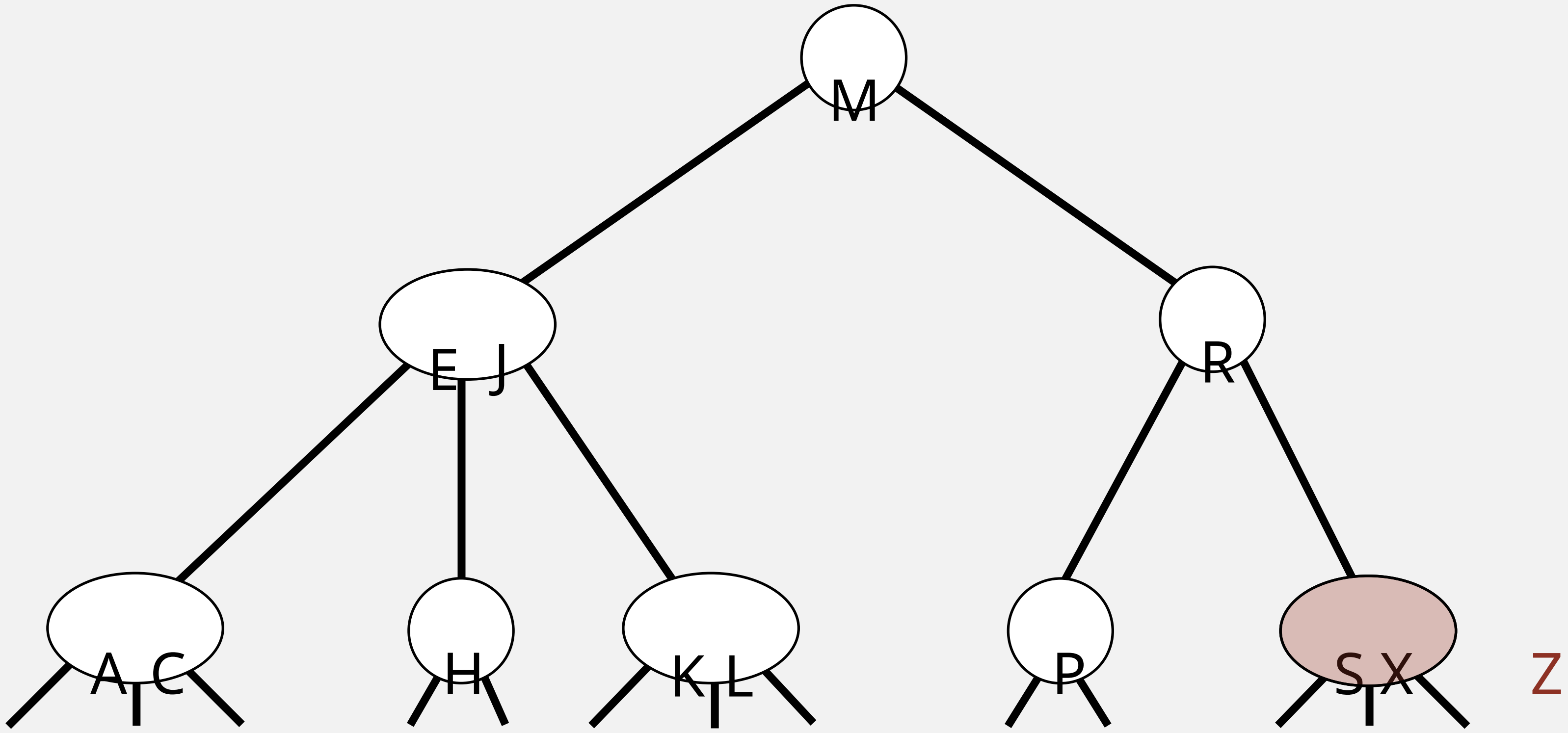
search ends here

2-3 tree demo: insertion

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



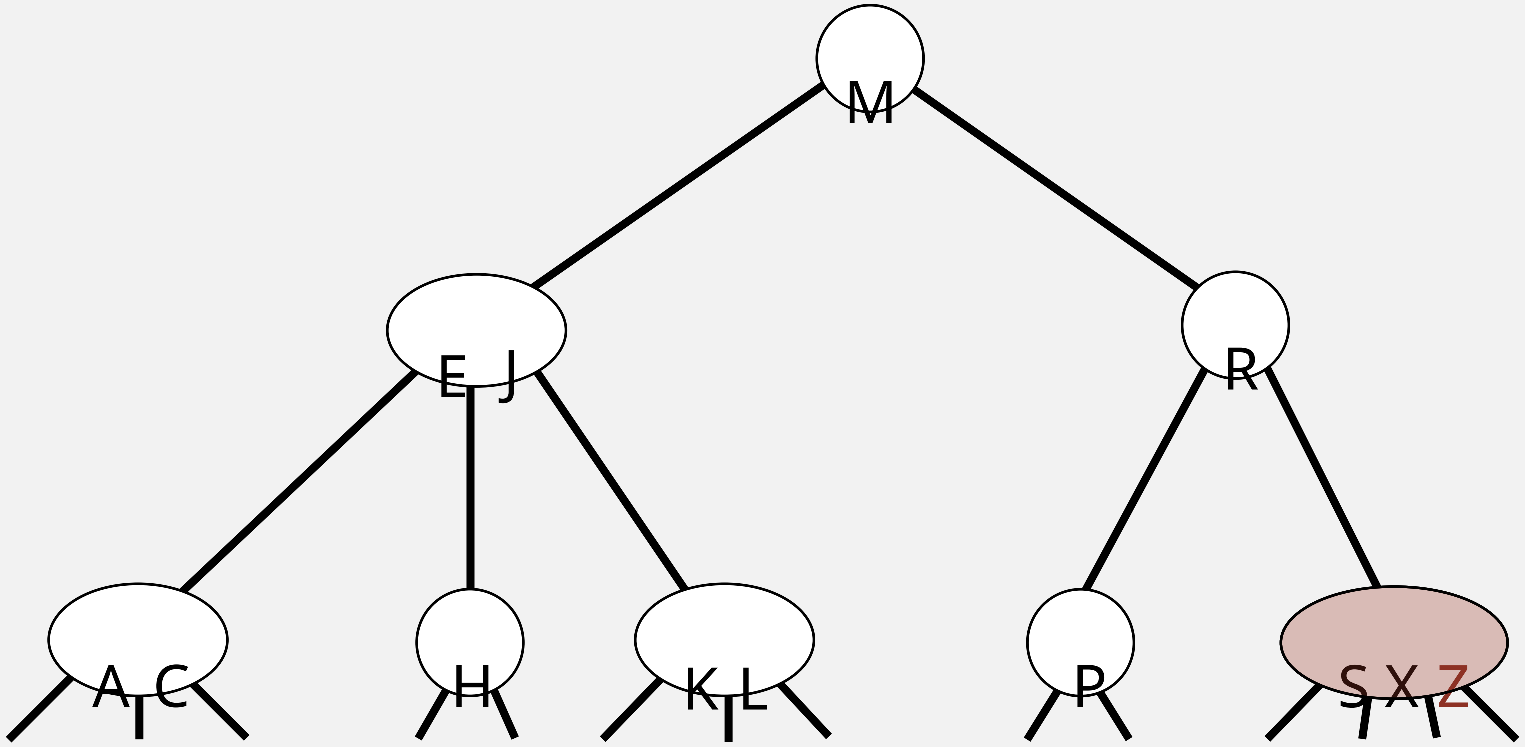
replace 3-node with
temporary 4-node containing Z

2-3 tree demo: insertion

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

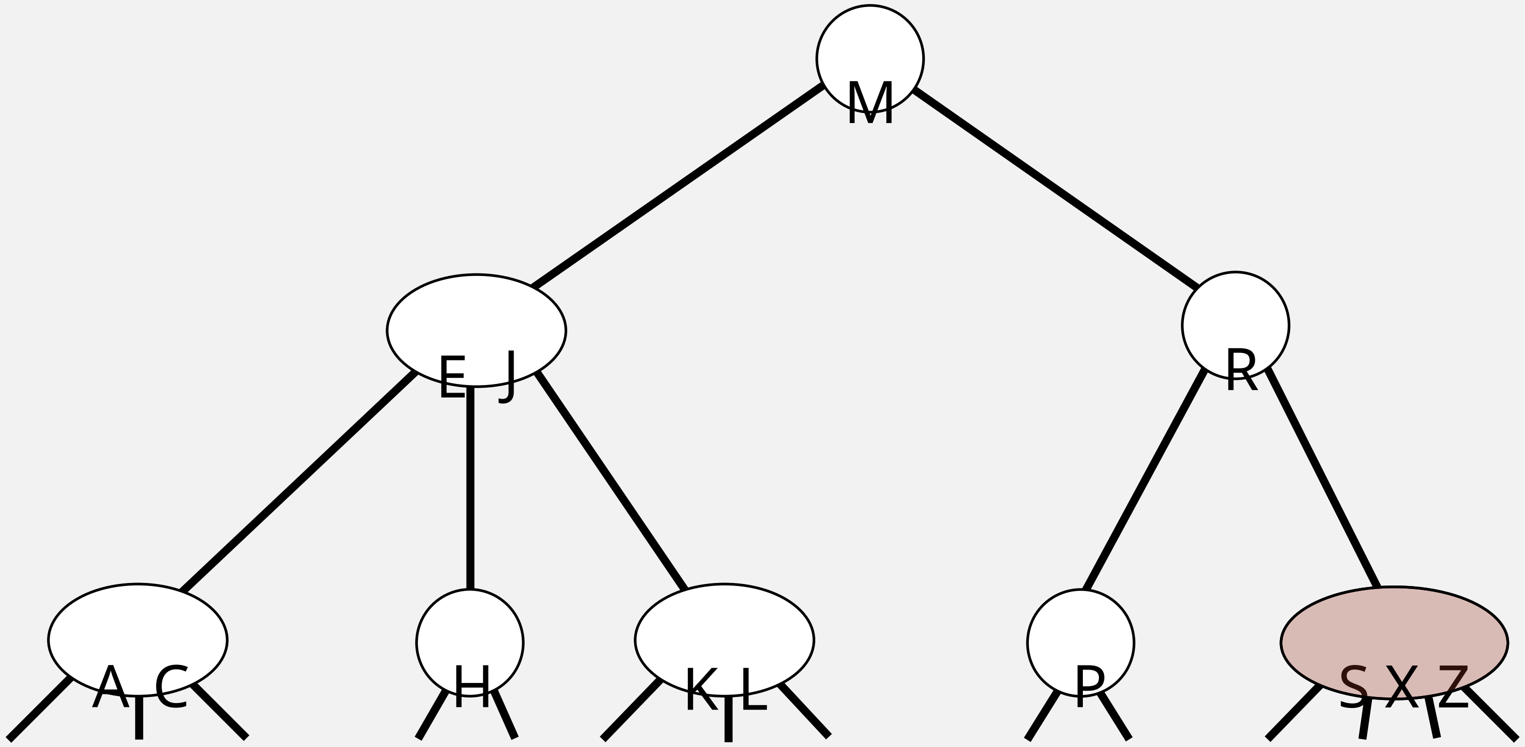


2-3 tree demo: insertion

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

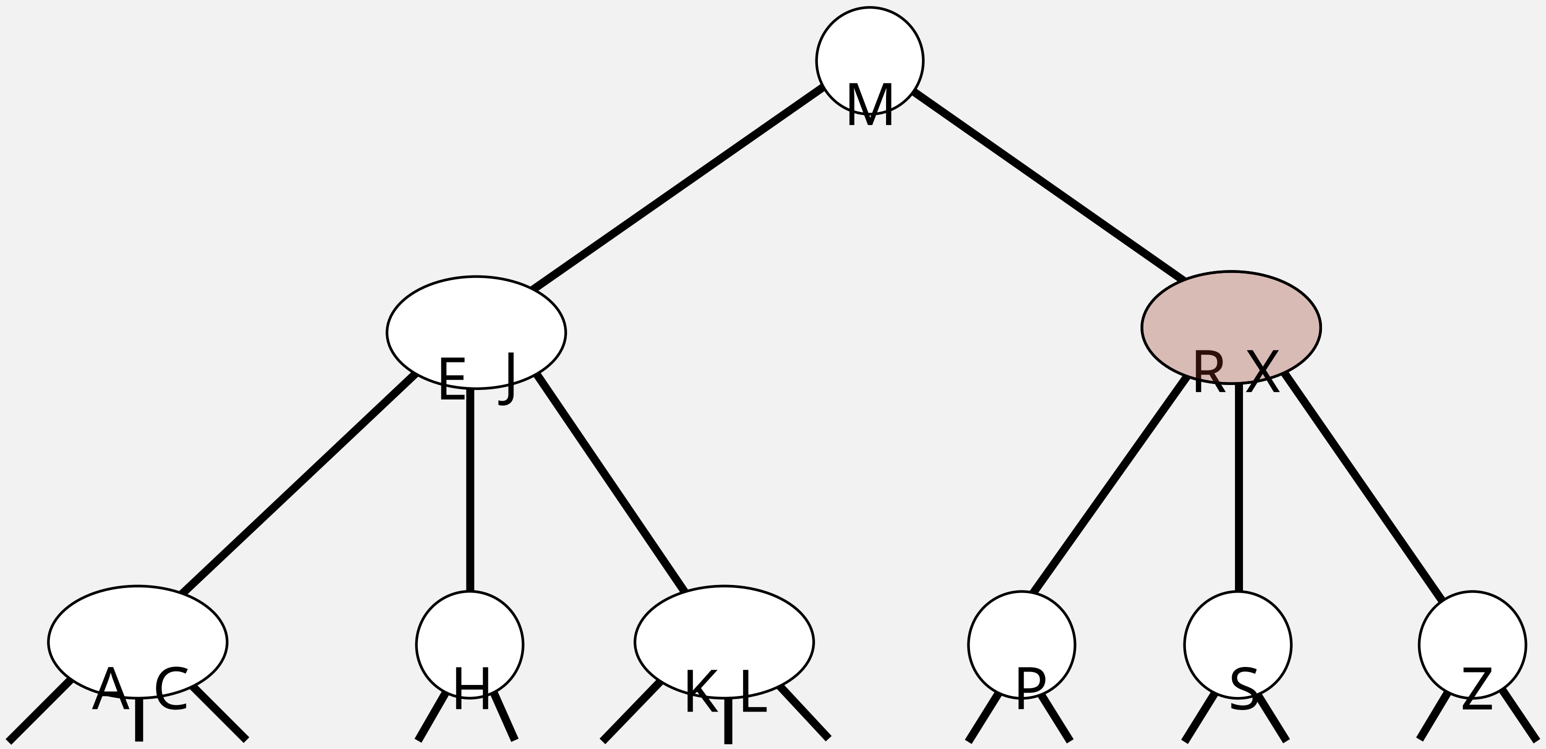
(pass middle key to parent)

2-3 tree demo: insertion

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

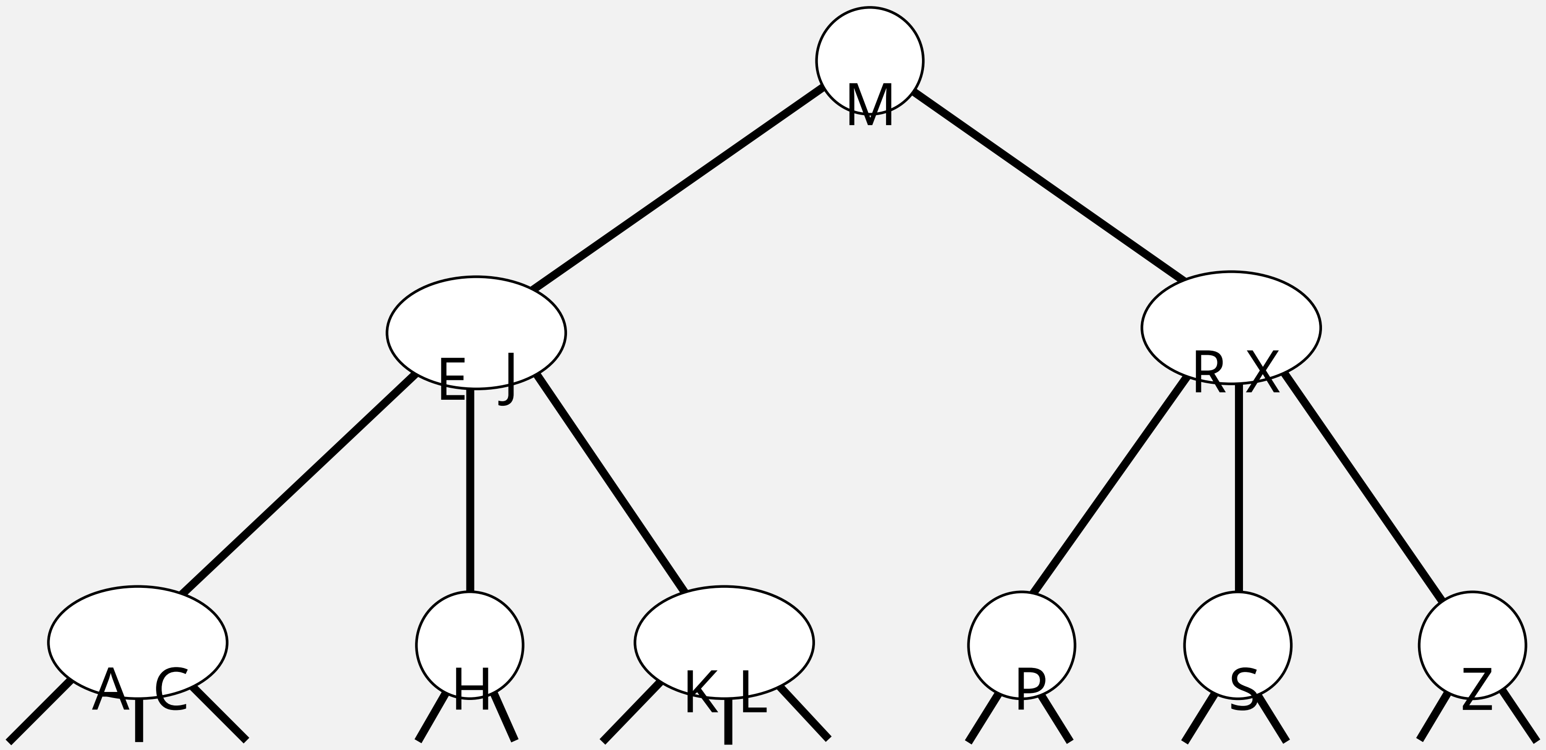


2-3 tree demo: insertion

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

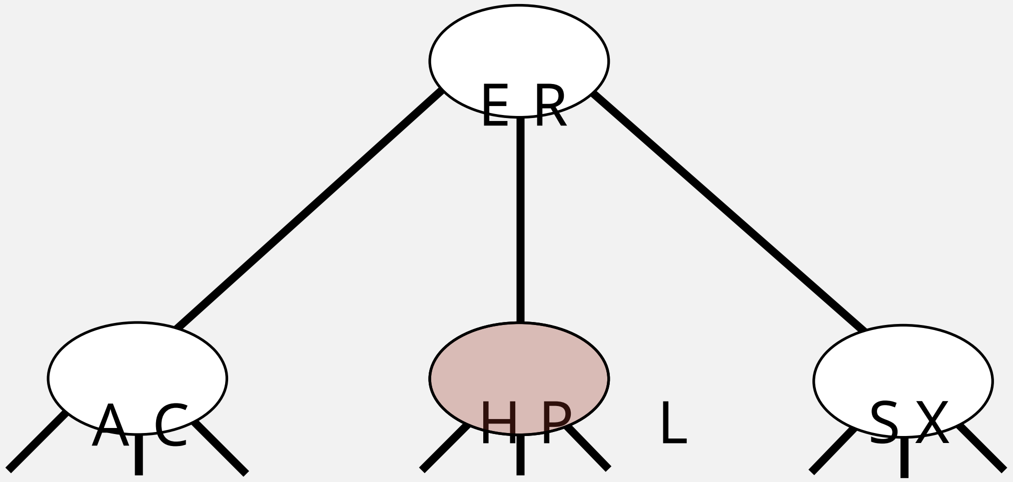


2-3 tree demo: insertion

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



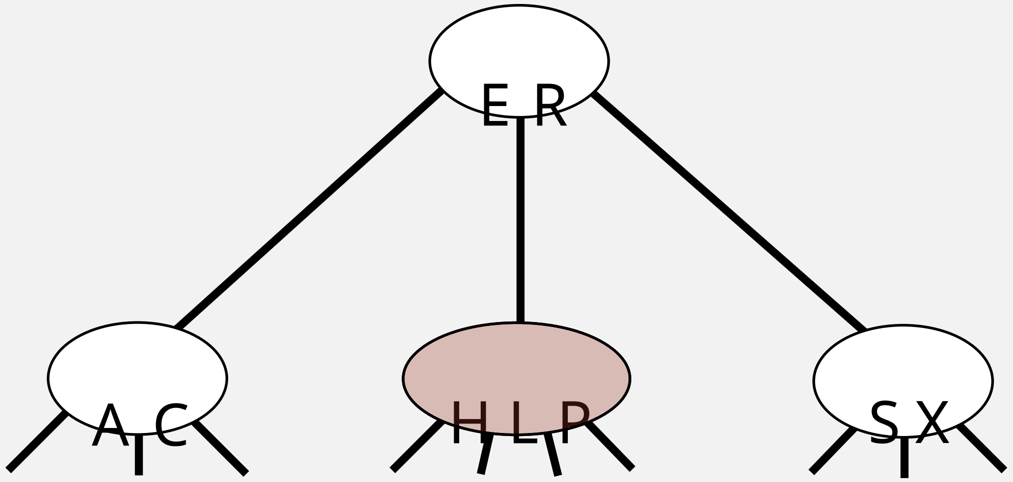
convert 3-node into 4-node

2-3 tree demo: insertion

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

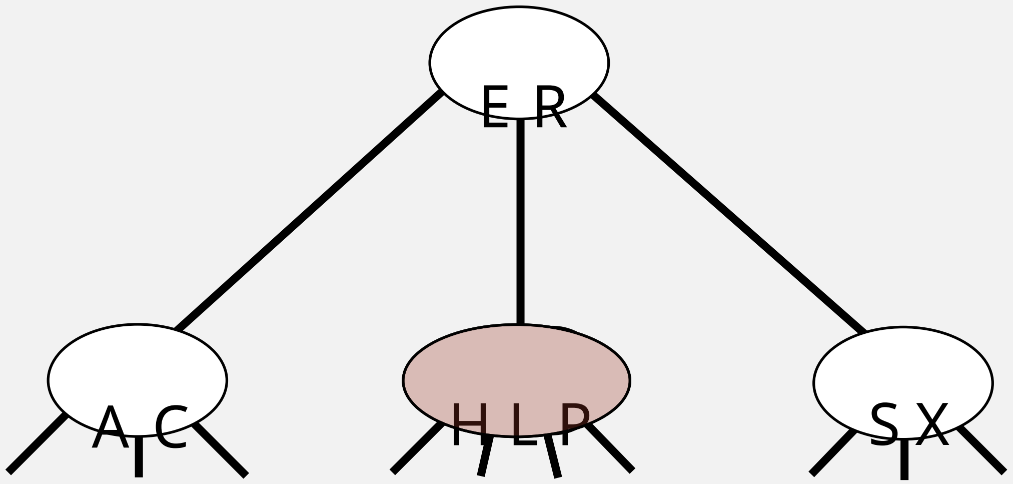


2-3 tree demo: insertion

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

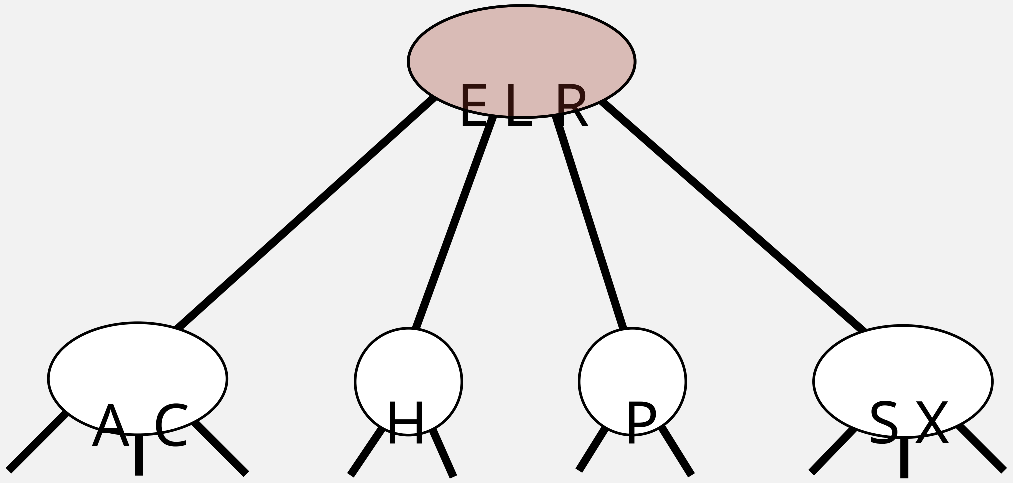
(move L to parent)

2-3 tree demo: insertion

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



2-3 tree demo: insertion

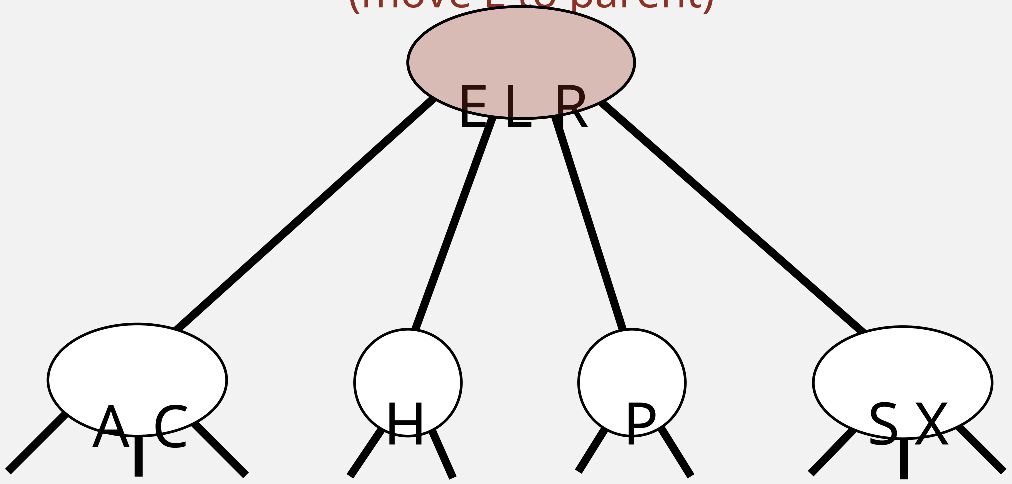
Insert into a 3-node at bottom.

- 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)



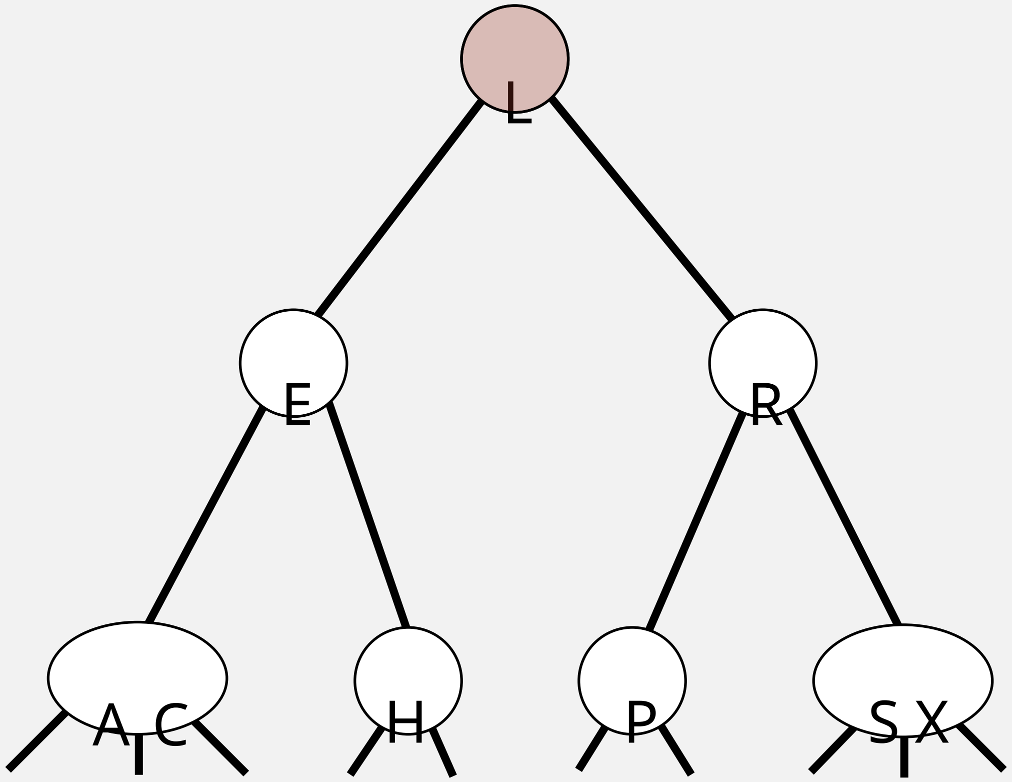
2-3 tree demo: insertion

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

height of tree increases by 1

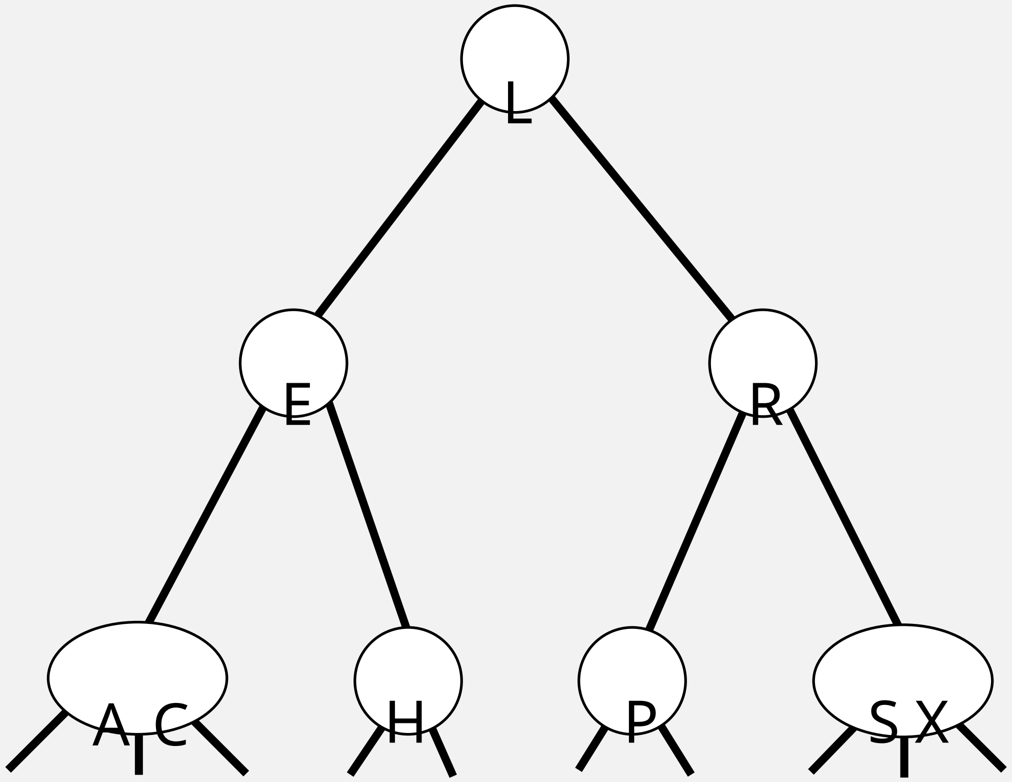


2-3 tree demo: insertion

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



Algorithms

Robert Sedgewick | Kevin Wayne

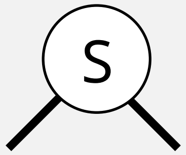
http://algs4.cs.princeton.edu

3.3 2-3 Tree Demo

- *search*
- *insertion*
- *construction*

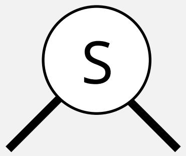
2-3 tree demo: construction

insert S



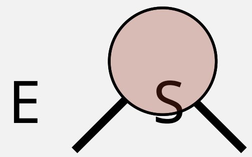
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

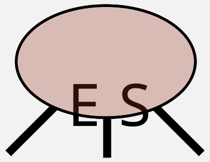
insert E



convert 2-node into 3-node

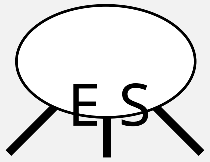
2-3 tree demo: construction

insert E



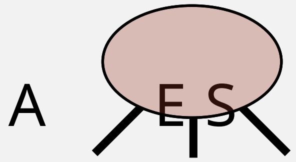
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

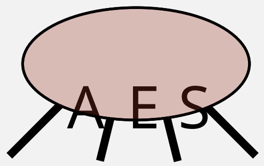
insert A



convert 3-node into 4-node

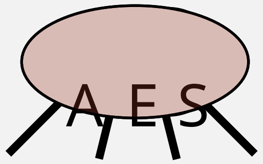
2-3 tree demo: construction

insert A



2-3 tree demo: construction

insert A

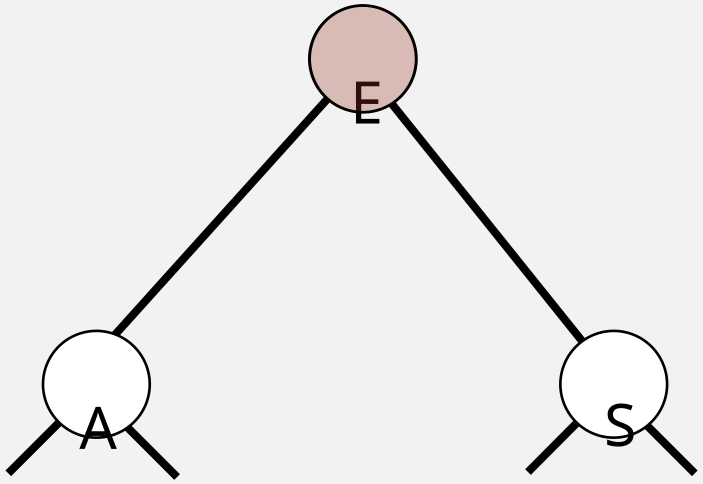


split 4-node

(move E to parent)

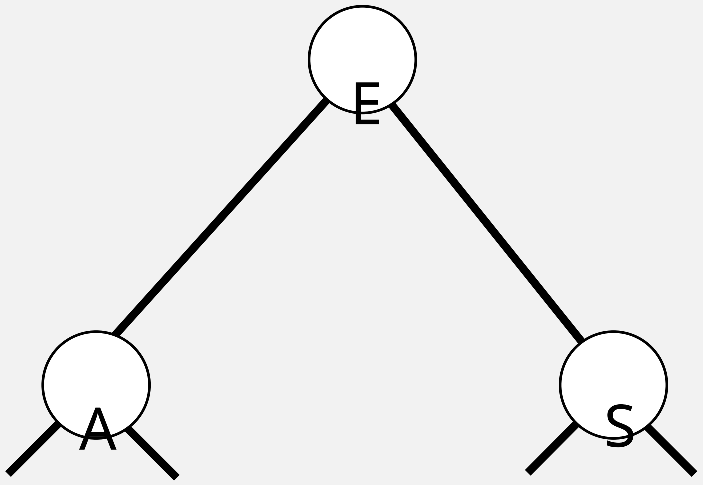
2-3 tree demo: construction

insert A



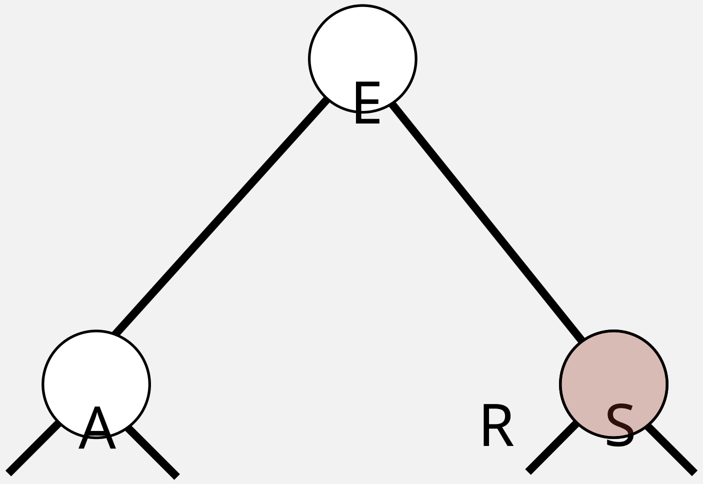
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

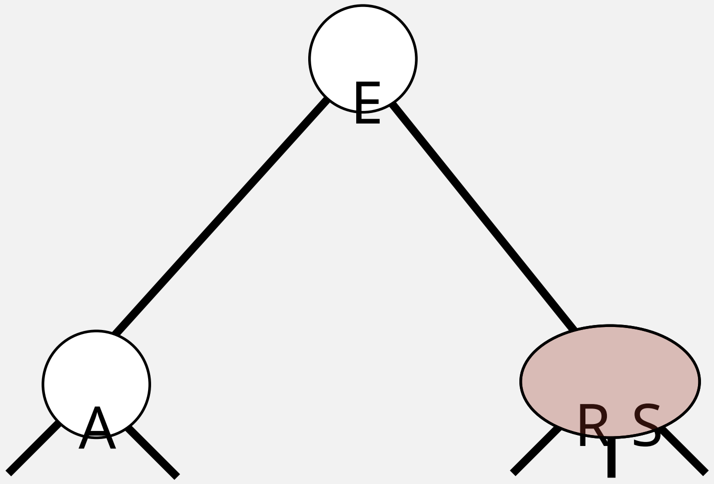
insert R



convert 2-node into 3-node

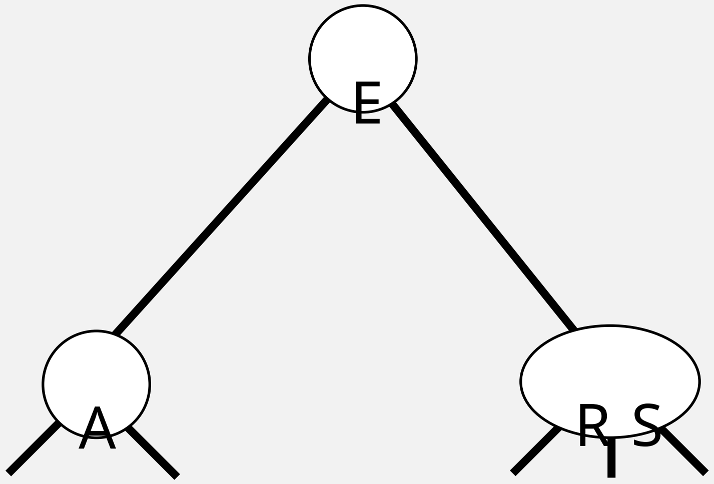
2-3 tree demo: construction

insert R



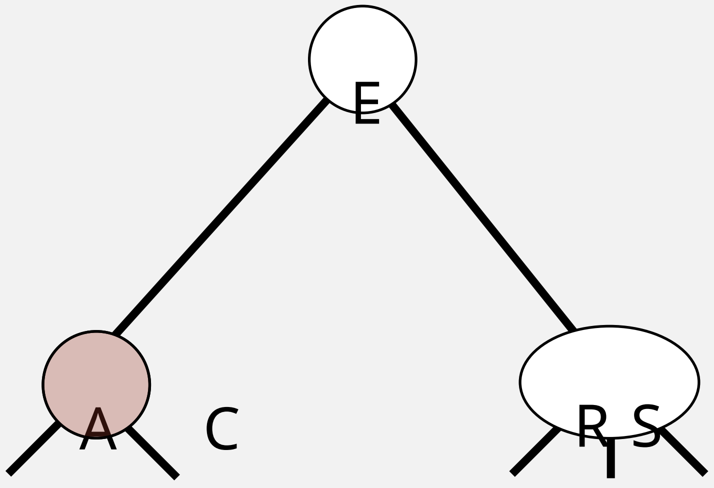
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

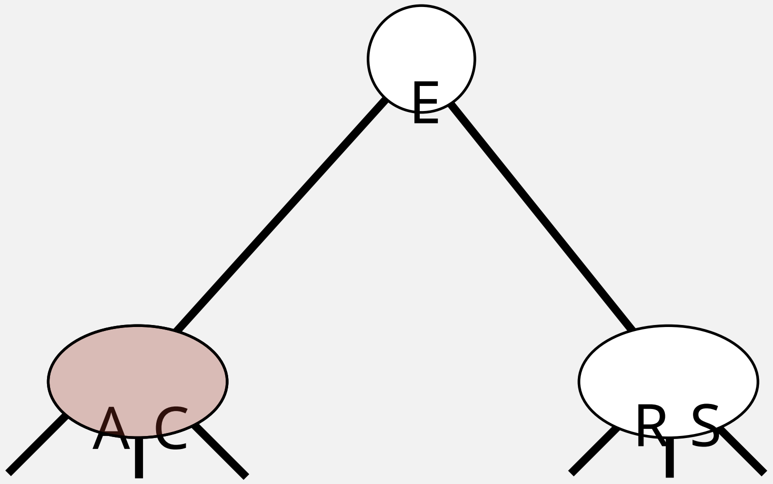
insert C



convert 2-node into 3-node

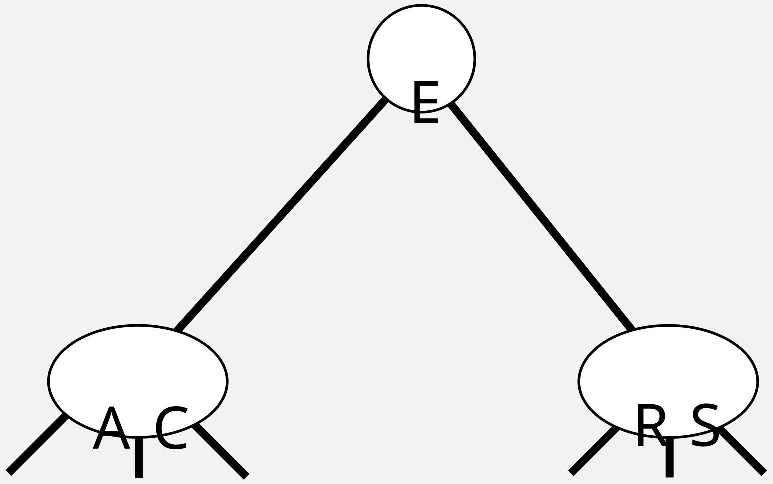
2-3 tree demo: construction

insert C



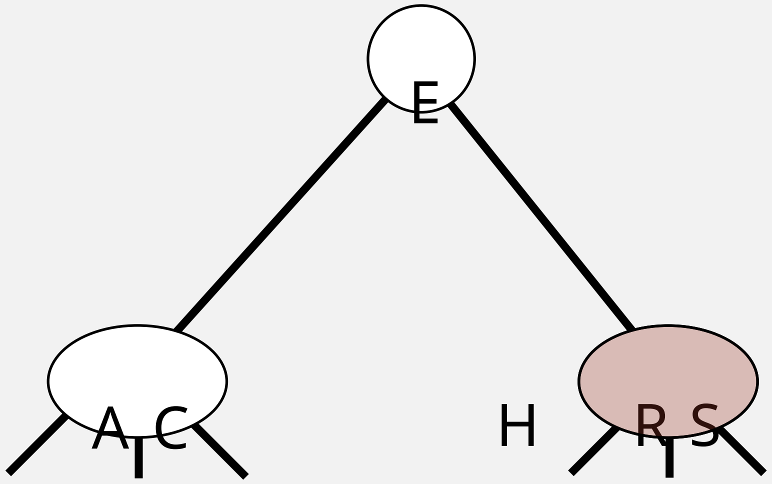
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

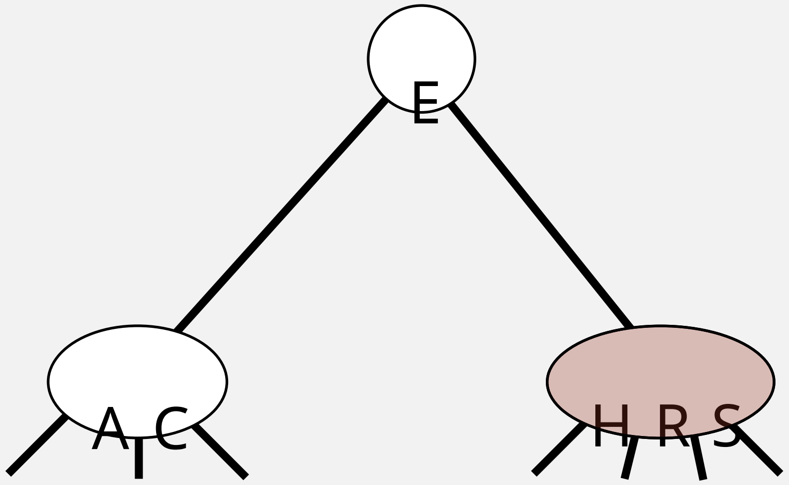
insert H



convert 3-node into 4-node

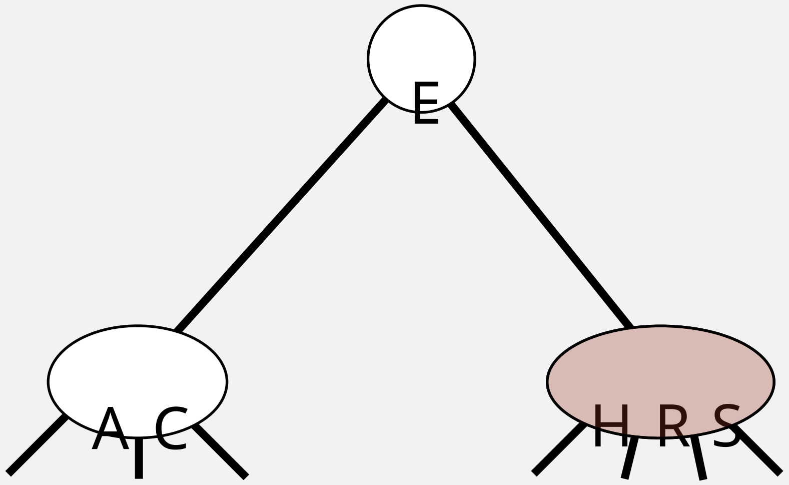
2-3 tree demo: construction

insert H



2-3 tree demo: construction

insert H

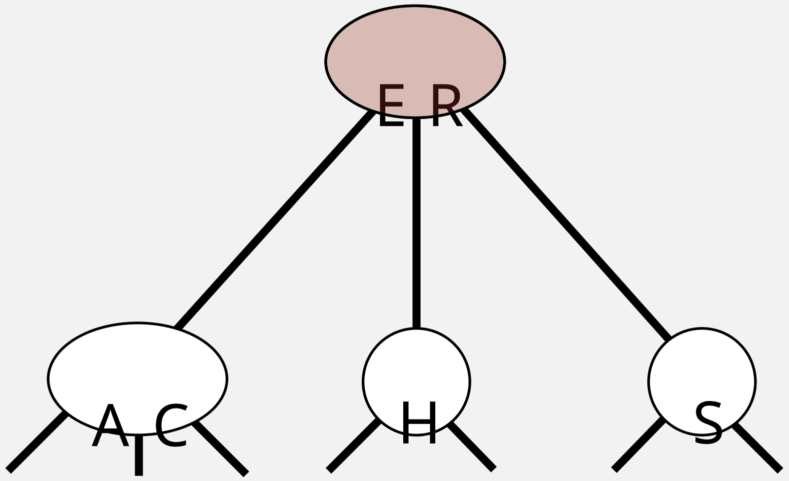


split 4-node

(move R to parent)

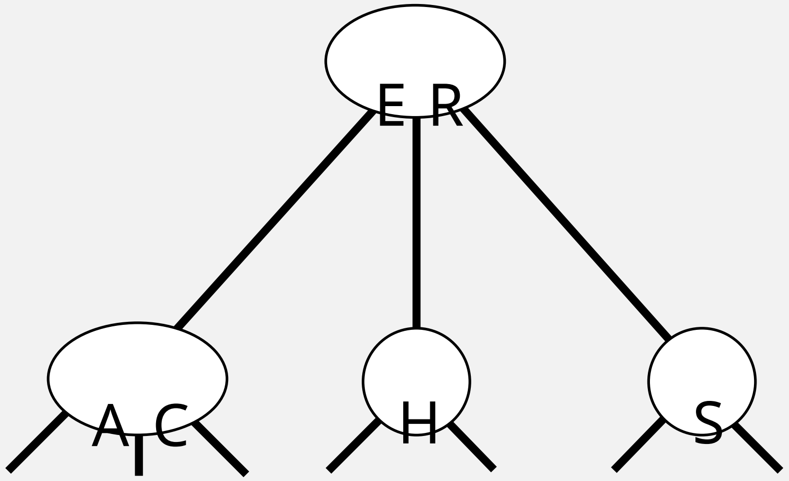
2-3 tree demo: construction

insert H



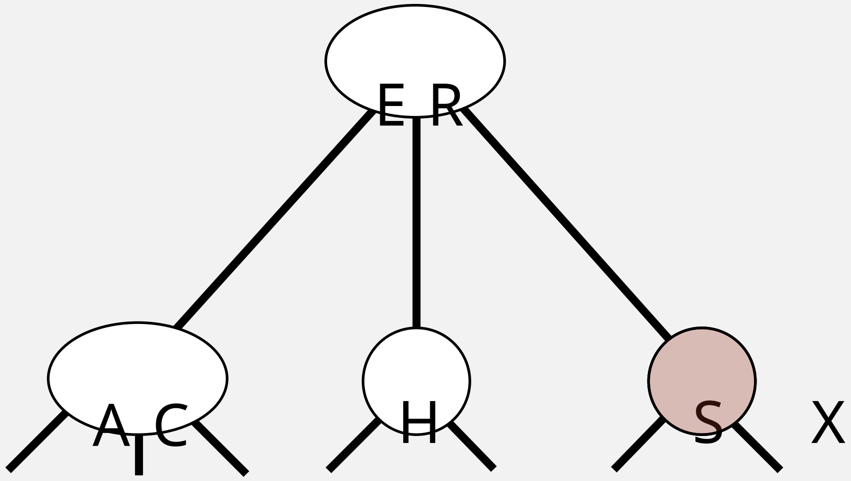
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

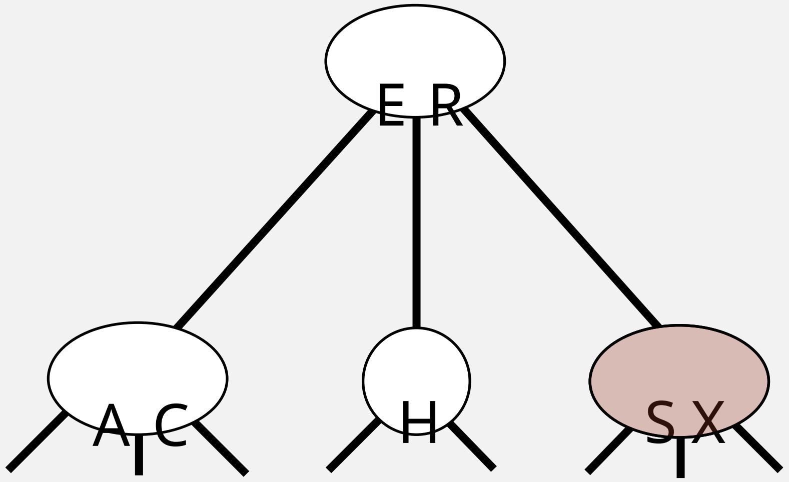
insert X



convert 2-node into 3-node

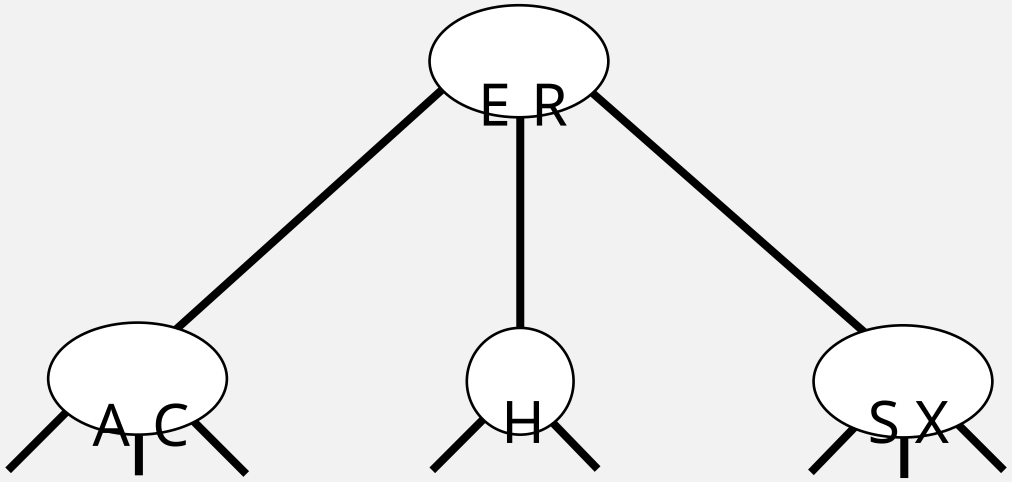
2-3 tree demo: construction

insert X



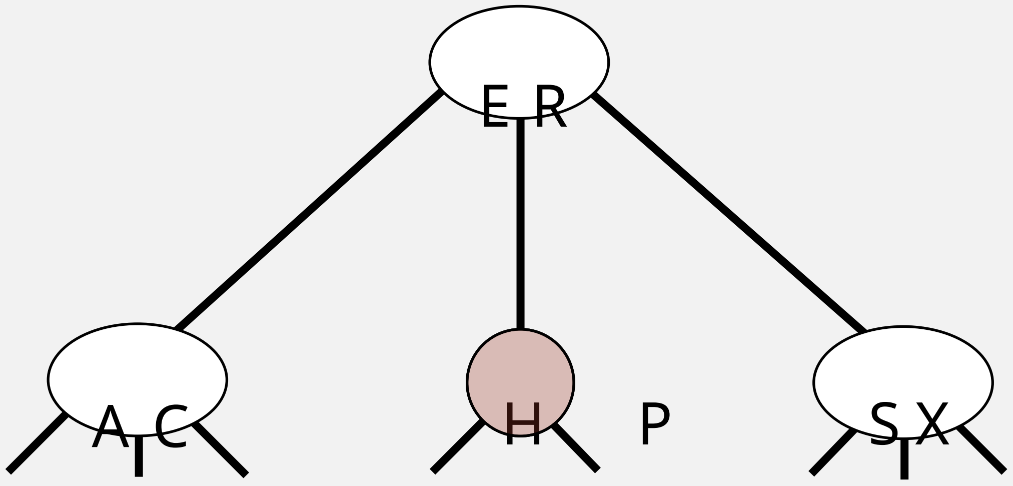
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

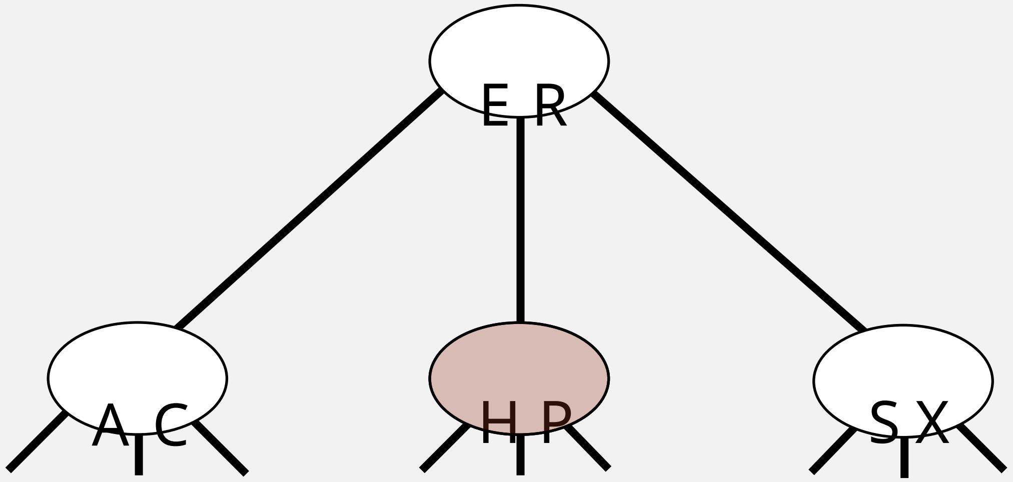
insert P



convert 2-node into 3-node

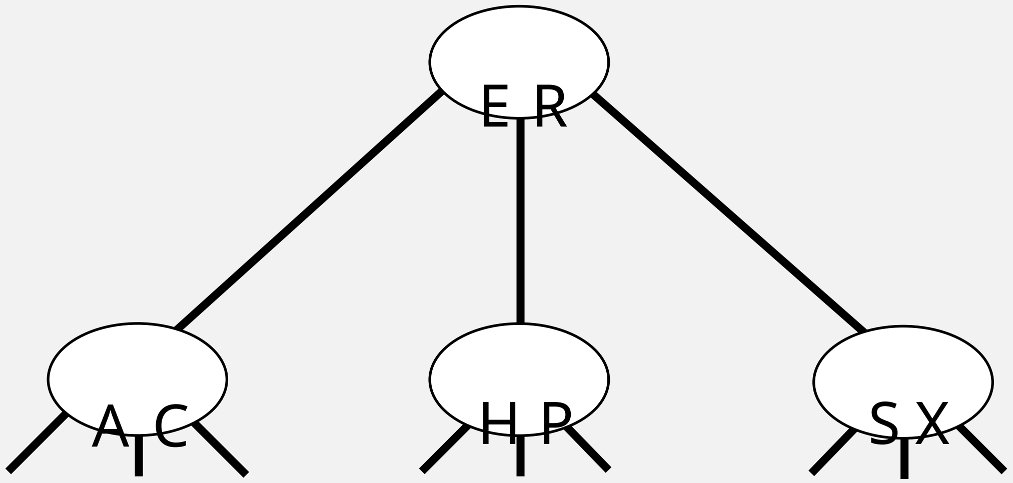
2-3 tree demo: construction

insert P



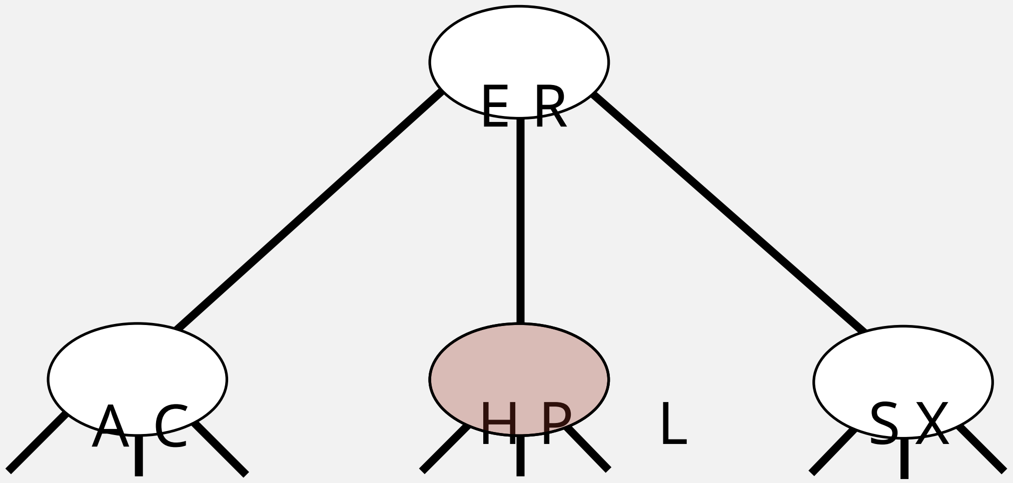
2-3 tree demo: construction

2-3 tree



2-3 tree demo: construction

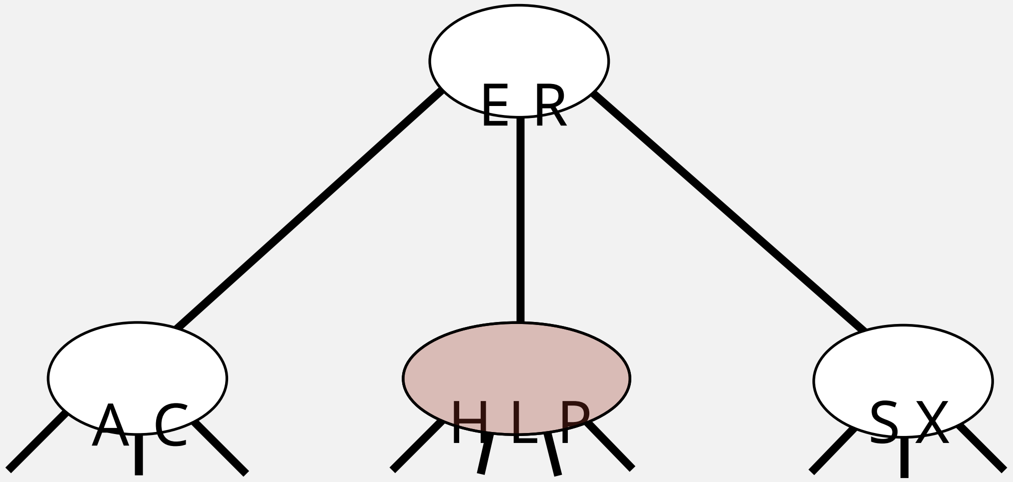
insert L



convert 3-node into 4-node

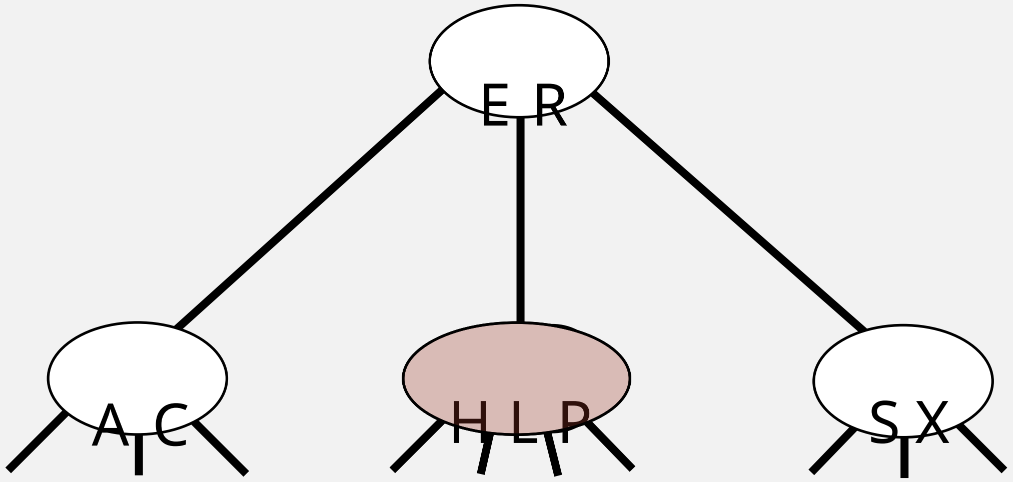
2-3 tree demo: construction

insert L



2-3 tree demo: construction

insert L

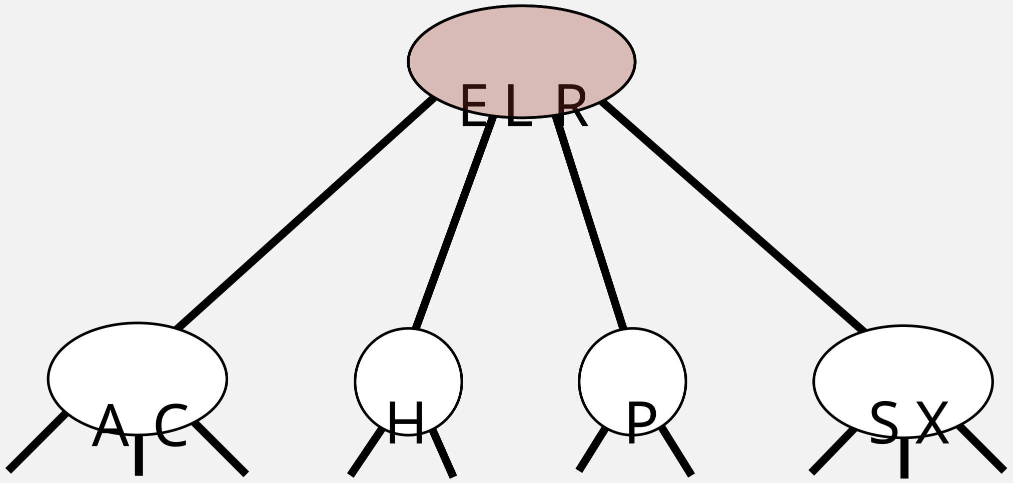


split 4-node

(move L to parent)

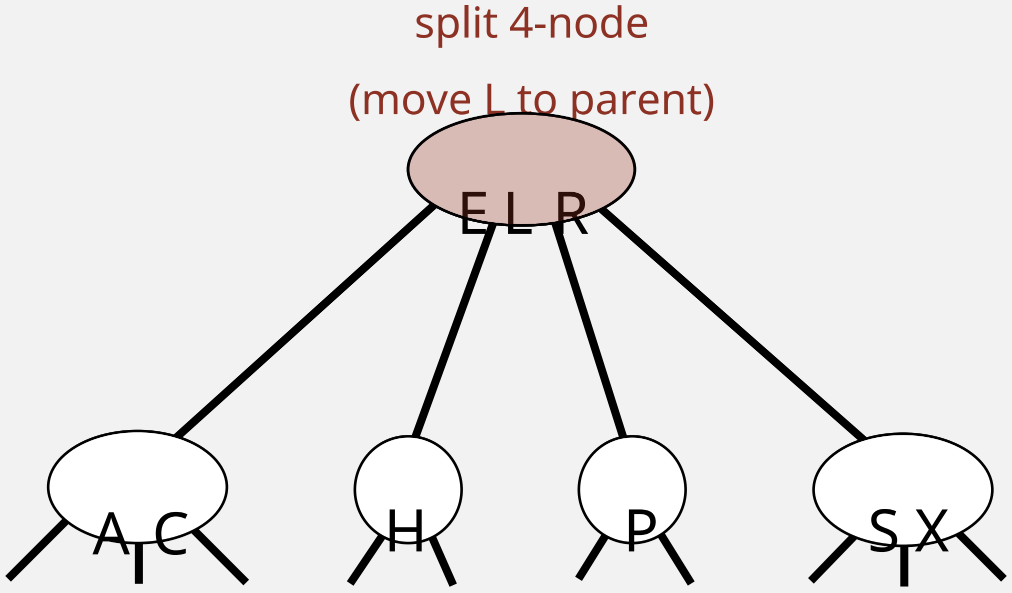
2-3 tree demo: construction

insert L



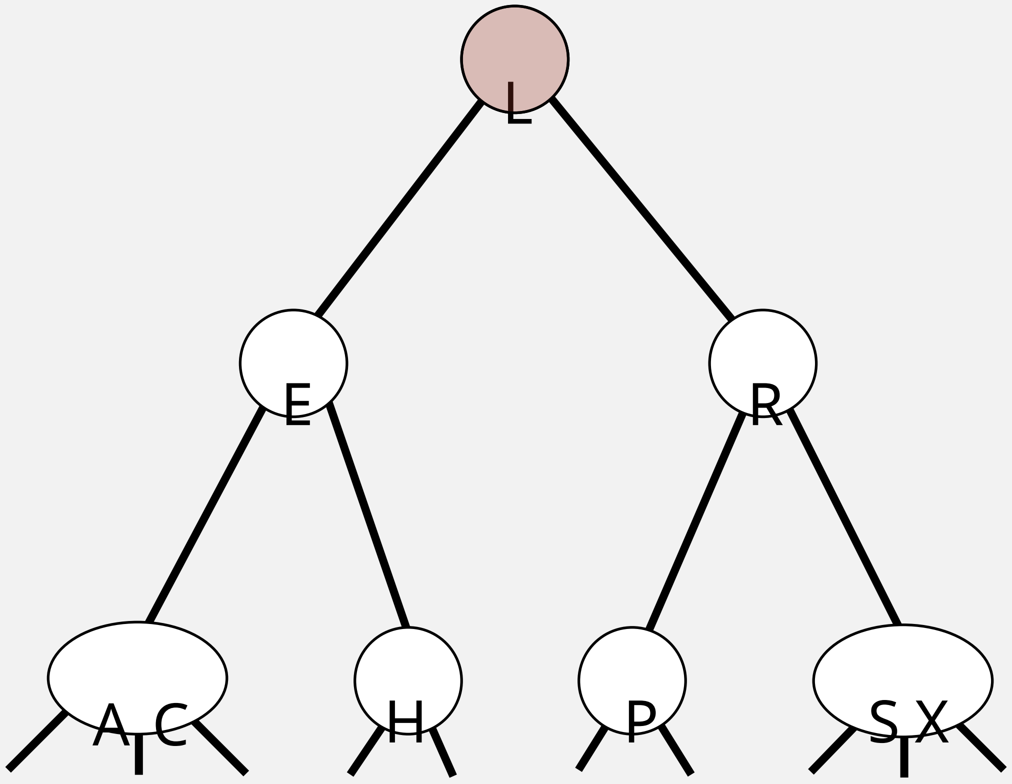
2-3 tree demo: construction

insert L



2-3 tree demo: construction

insert L



2-3 tree demo: construction

2-3 tree