

CSc 3102: B-Trees

Intro, Insertion, Split and Traversals

Drawn by Rohan Kadkol based on Dr. Duncan's handout

Lecture # 23: §4.7

- ❑ Background
- ❑ Some Applications
- ❑ Properties of B-trees
- ❑ B-tree Insertion
- ❑ B-tree Traversal
- ❑ Non-graphic B-tree Representation

B-Trees

- m-th order B-tree
- m here is the order of the B-tree
- Order:
 - At most how many **children**
 - $\# \text{ keys} = \# \text{ children} - 1 = \text{order} - 1$

B-Trees

- Max # children = order = m
- Max # keys = order - 1 = $m - 1$
- Min # of children = $\lceil m/2 \rceil$
- Min # of keys for **non root** = $\lceil m/2 \rceil - 1$
- # keys for **root**:
 - No children, or
 - 2 or more children (the min doesn't depend on m)

B-Trees Operations

- Insert
 - Split
 - The middle key is called the median key
 - To determine the median when m is even,
 - Median index = $(m-1) / 2$ (whether even or odd as we use integer division)
- Delete
 - Merge

B-Tree Insertion Example

B-Trees Insertion and Split

Q) Insert the following nodes in order in a 5th order B-Tree: N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split

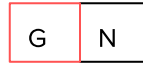
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split

N

N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split

B	G	N
---	---	---

N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split

A	B	G	N
---	---	---	---

N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split

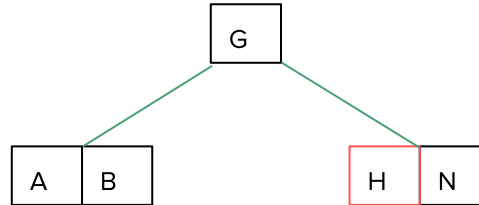
A	B	G	H	N
---	---	---	---	---

keys = 5 > 5 - 1 (m-1)

Hence, split

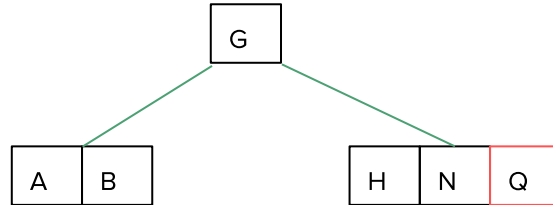
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



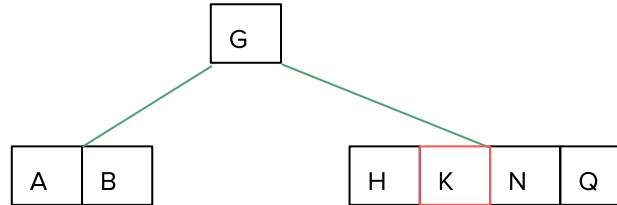
N, G, B, A, **H**, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



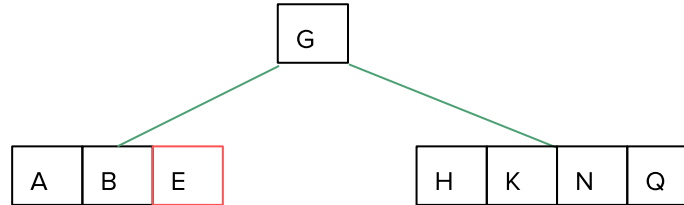
N, G, B, A, H, **Q**, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



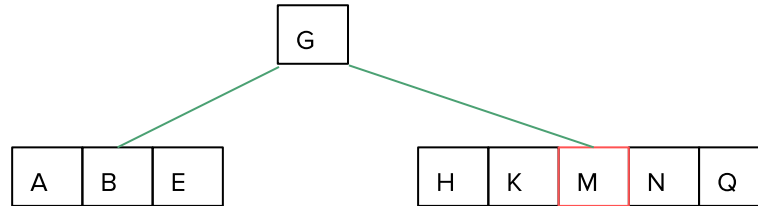
N, G, B, A, H, Q, **K**, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



N, G, B, A, H, Q, K, **E**, M, T, V, F, L, Z, D, X, Y, R, P, and S

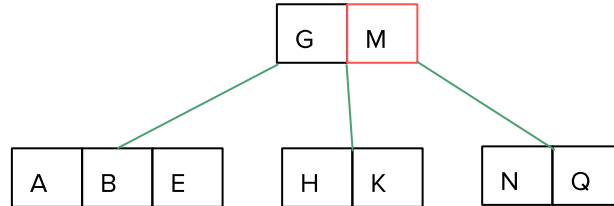
B-Trees Insertion and Split



keys = 5 > 5 - 1 (m-1)
Hence, split

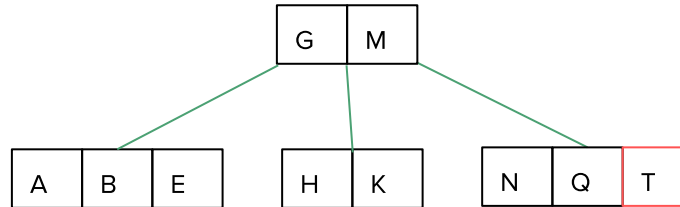
N, G, B, A, H, Q, K, E, **M**, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



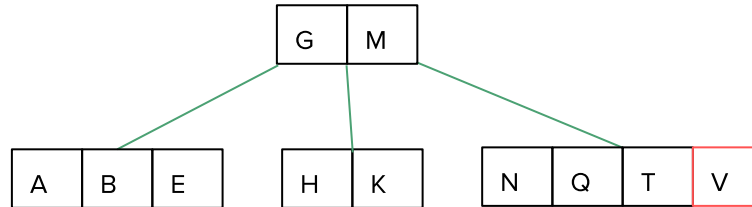
N, G, B, A, H, Q, K, E, **M**, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



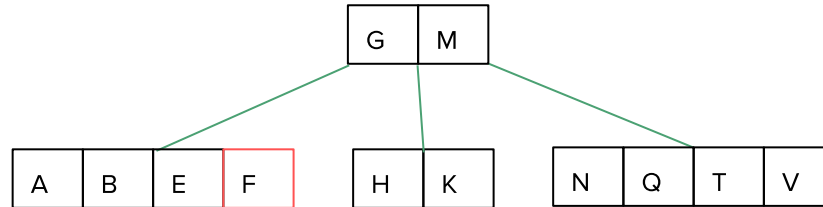
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



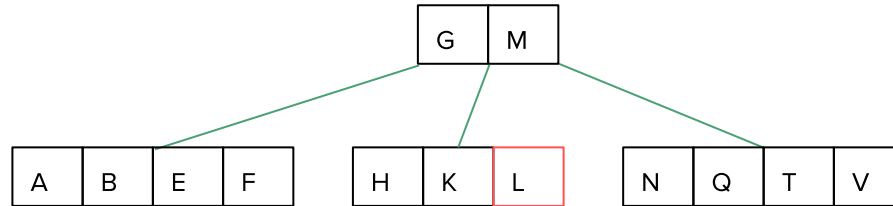
N, G, B, A, H, Q, K, E, M, T, **V**, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



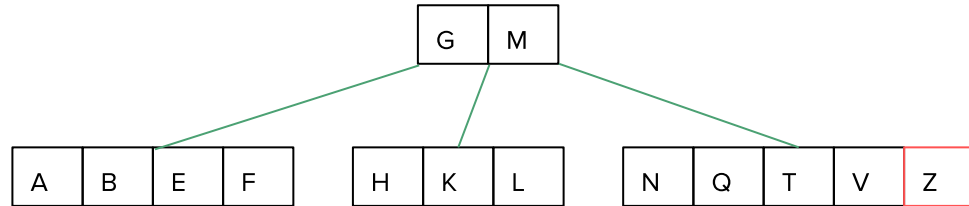
N, G, B, A, H, Q, K, E, M, T, V, **F**, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



N, G, B, A, H, Q, K, E, M, T, V, F, **L**, Z, D, X, Y, R, P, and S

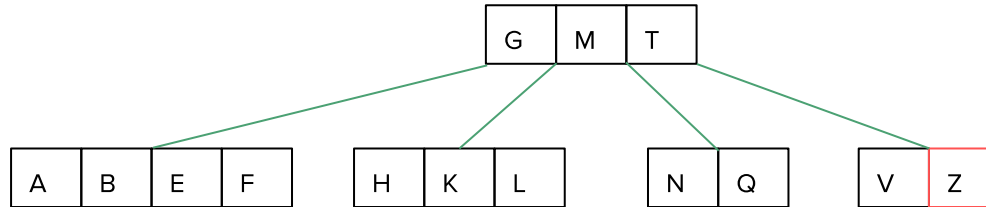
B-Trees Insertion and Split



keys = 5 > 5 - 1 (m-1)
Hence, split

N, G, B, A, H, Q, K, E, M, T, V, F, L, **Z**, D, X, Y, R, P, and S

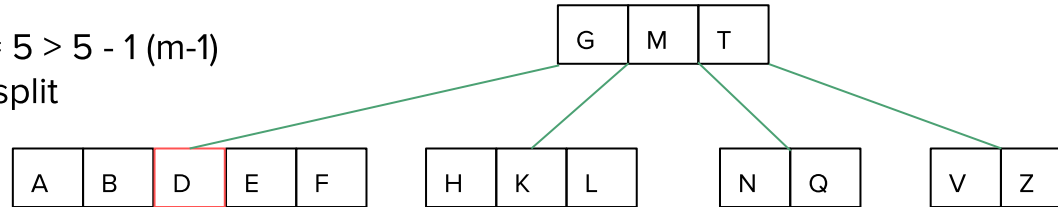
B-Trees Insertion and Split



N, G, B, A, H, Q, K, E, M, T, V, F, L, **Z**, D, X, Y, R, P, and S

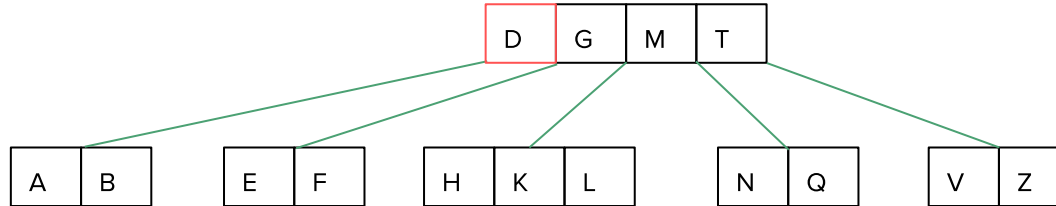
B-Trees Insertion and Split

keys = 5 > 5 - 1 (m-1)
Hence, split



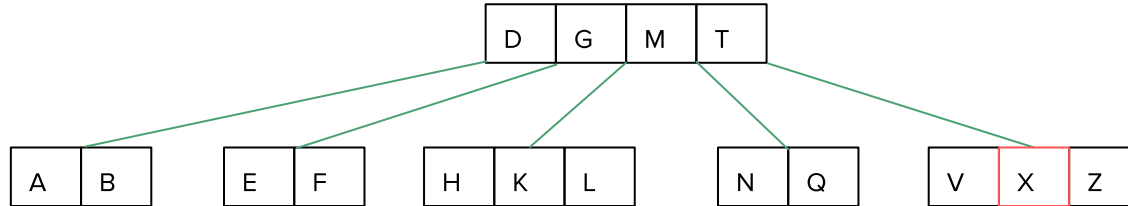
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, **D**, X, Y, R, P, and S

B-Trees Insertion and Split



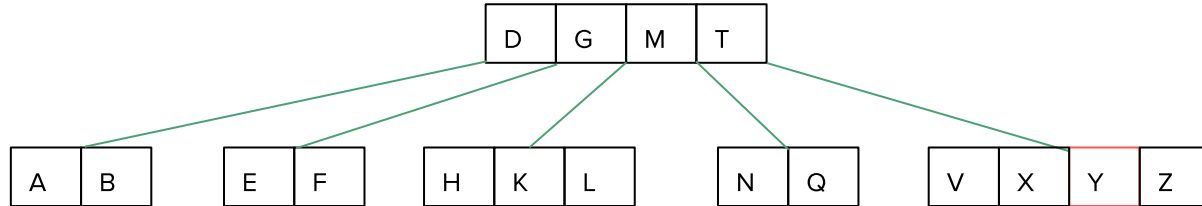
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, **D**, X, Y, R, P, and S

B-Trees Insertion and Split



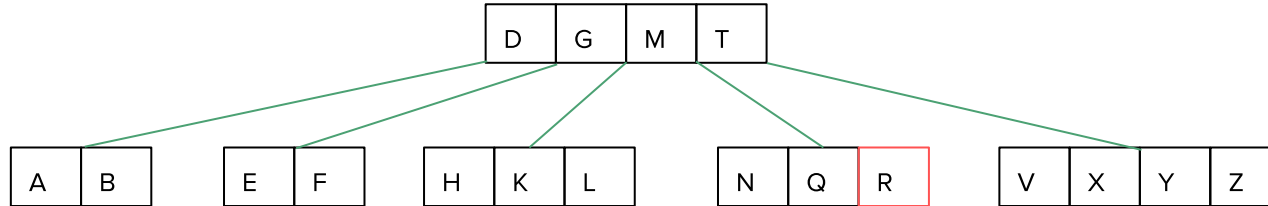
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, **X**, Y, R, P, and S

B-Trees Insertion and Split



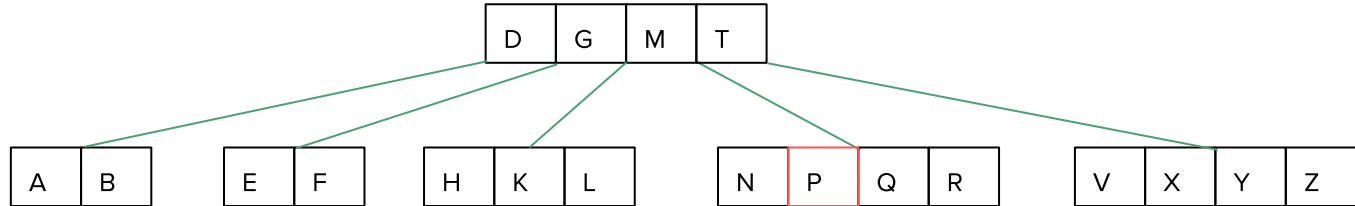
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



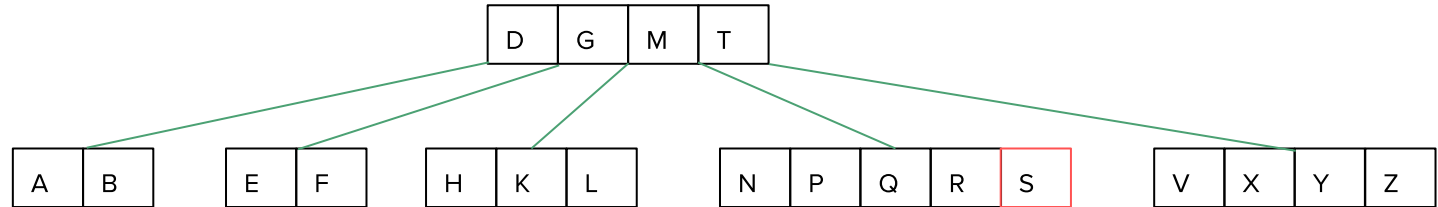
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, **P**, and S

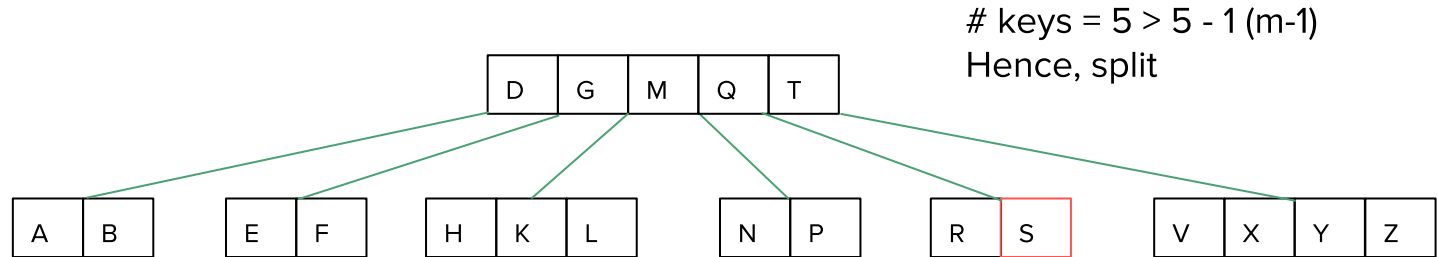
B-Trees Insertion and Split



keys = 5 > 5 - 1 (m-1)
Hence, split

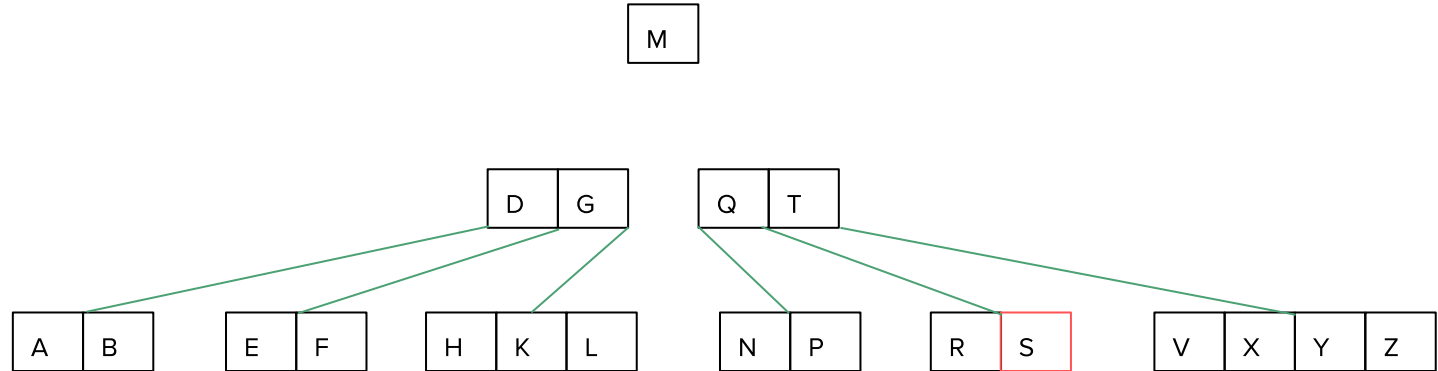
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Trees Insertion and Split



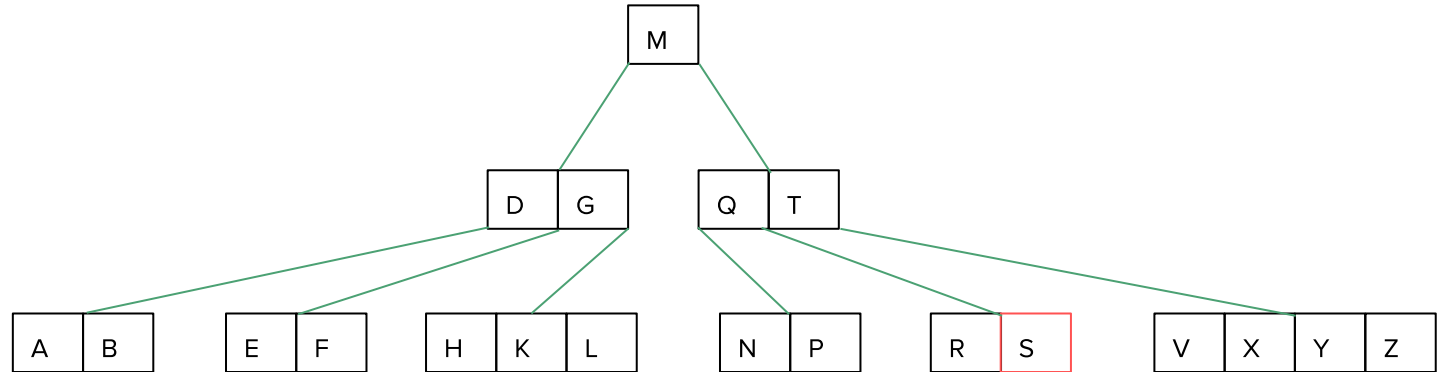
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and **S**

B-Trees Insertion and Split



N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

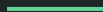
B-Trees Insertion and Split



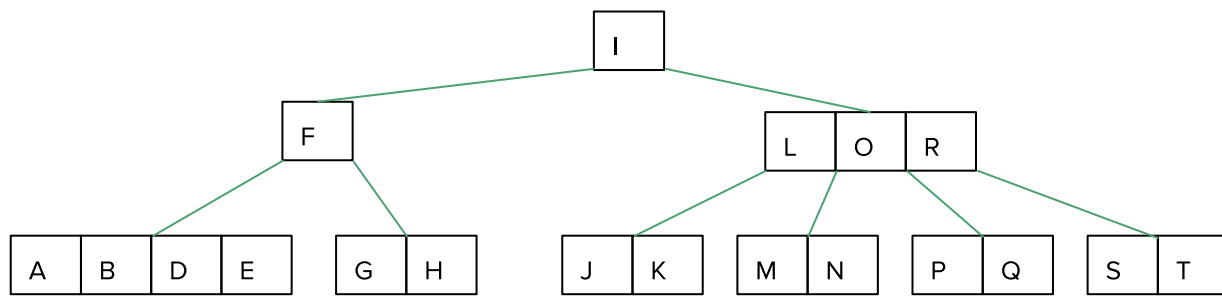
N, G, B, A, H, Q, K, E, M, T, V, F, L, Z, D, X, Y, R, P, and S

B-Tree Traversals

- In-order traversal
- Pre-order traversal
- Post-order traversal



In-Order Traversal



Algorithm::In-Order

Interleave the in-order and visit calls between the current node's entries and child nodes:

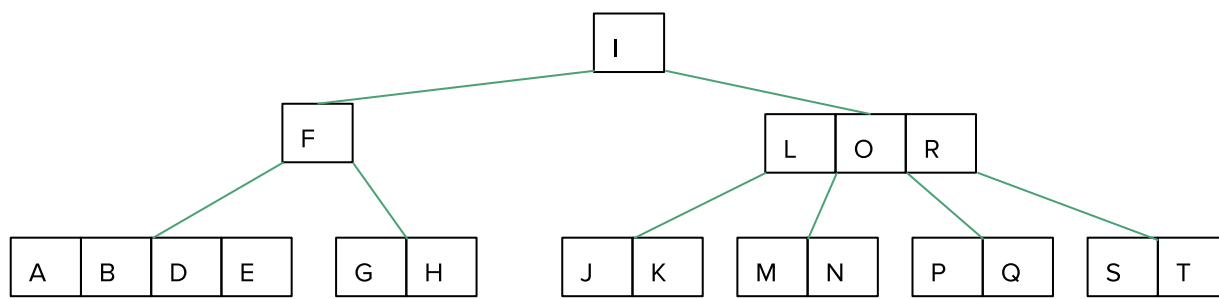
currentNodeEntries[$e_1 \dots e_n$] and currentNodeChildren[$c_1 \dots c_{n+1}$]

For $i=1:n$

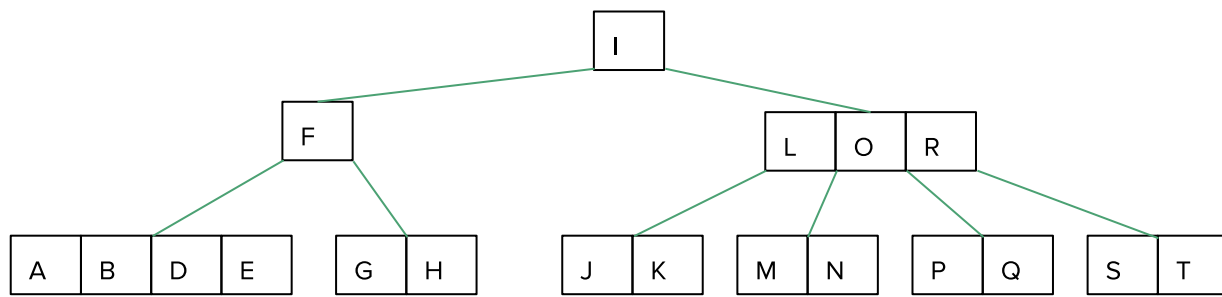
 In-Order (currentNodeChildren[c_i]))

 Visit(currentNodeEntries(e_i))

 In-Order (currentNodeChildren[c_{n+1}]))

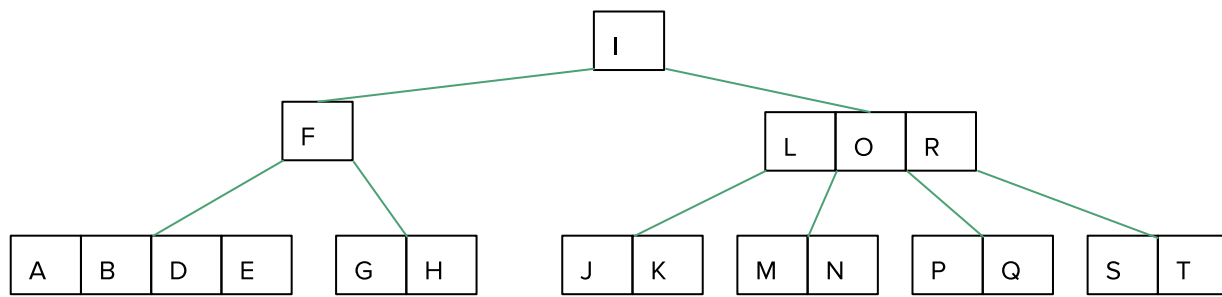


In-Order ([F]) I In-Order([LOR])



In-Order ([F]) I In-Order ([LOR])

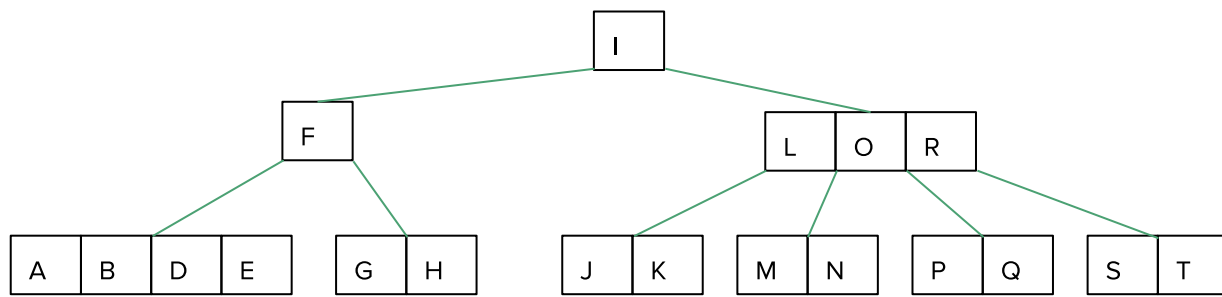
In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])



In-Order ([F]) I In-Order ([LOR])

In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

A B D E F In-Order ([GH]) I In-Order ([LOR])

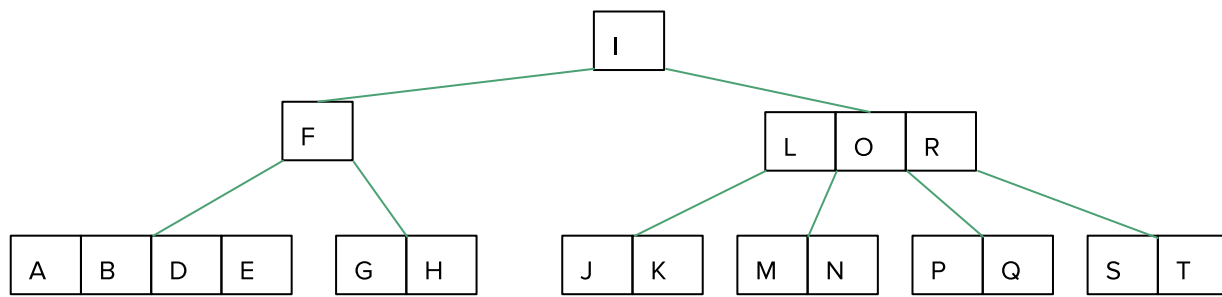


In-Order ([F]) I In-Order ([LOR])

In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

A B D E F In-Order ([GH]) I In-Order ([LOR])

A B D E F G H I In-Order ([LOR])



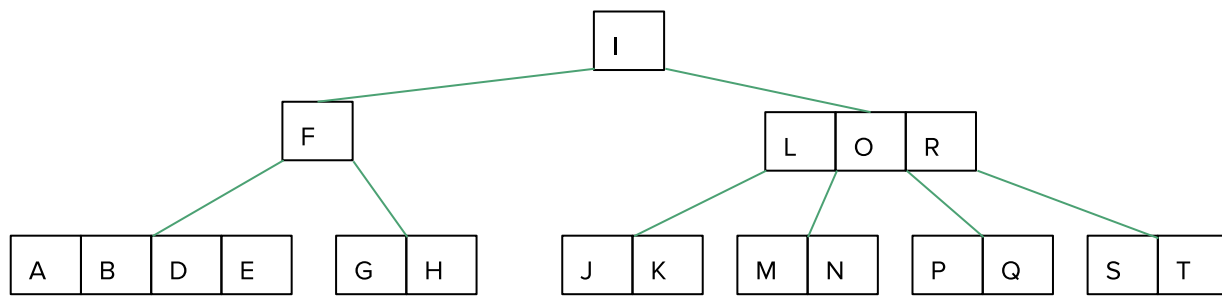
In-Order ([F]) I In-Order ([LOR])

In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

A B D E F In-Order ([GH]) I In-Order ([LOR])

A B D E F G H I In-Order ([LOR])

A B D E F G H I In-Order ([JK]) L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])



In-Order ([F]) I In-Order ([LOR])

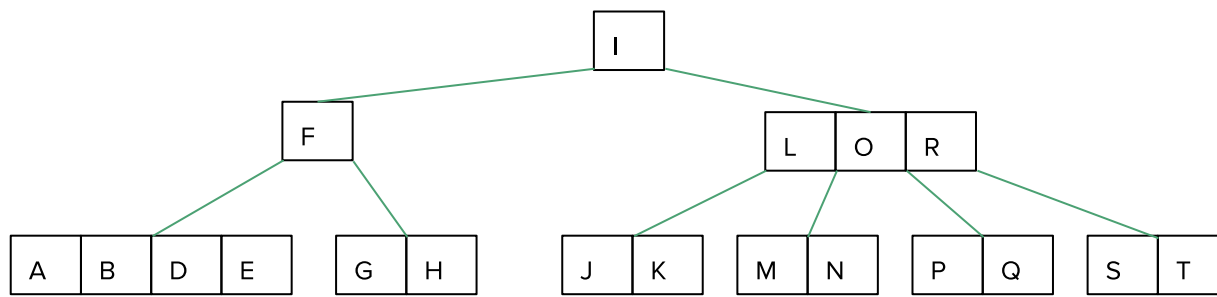
In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

A B D E F In-Order ([GH]) I In-Order ([LOR])

A B D E F G H I In-Order ([LOR])

A B D E F G H I In-Order ([JK]) L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])



In-Order ([F]) I In-Order ([LOR])

In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

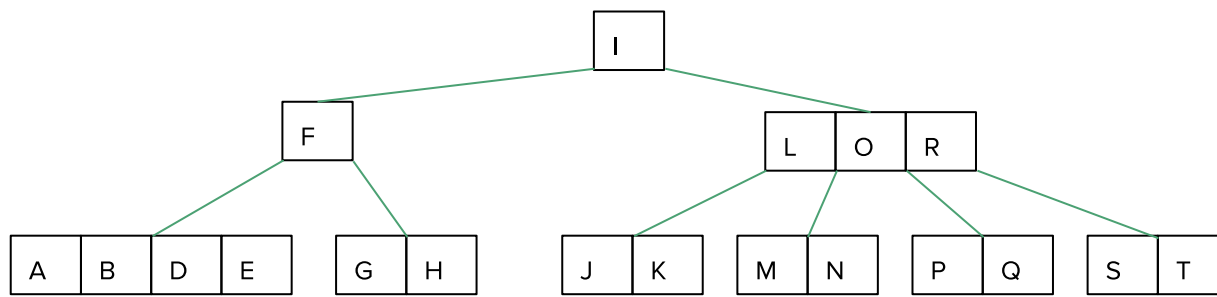
A B D E F In-Order ([GH]) I In-Order ([LOR])

A B D E F G H I In-Order ([LOR])

A B D E F G H I In-Order ([JK]) L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L M N O In-Order ([PQ]) R In-Order ([ST])



In-Order ([F]) I In-Order ([LOR])

In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

A B D E F In-Order ([GH]) I In-Order ([LOR])

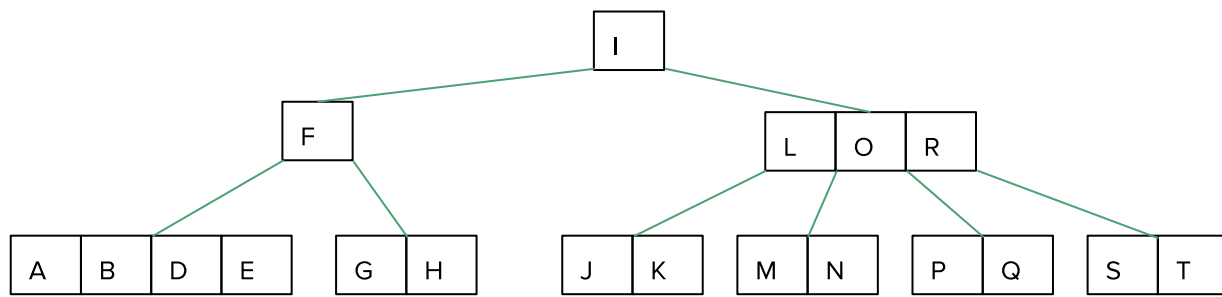
A B D E F G H I In-Order ([LOR])

A B D E F G H I In-Order ([JK]) L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L M N O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L M N O P Q R In-Order ([ST])



In-Order ([F]) I In-Order ([LOR])

In-Order ([ABDE]) F In-Order ([GH]) I In-Order ([LOR])

A B D E F In-Order ([GH]) I In-Order ([LOR])

A B D E F G H I In-Order ([LOR])

A B D E F G H I In-Order ([JK]) L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

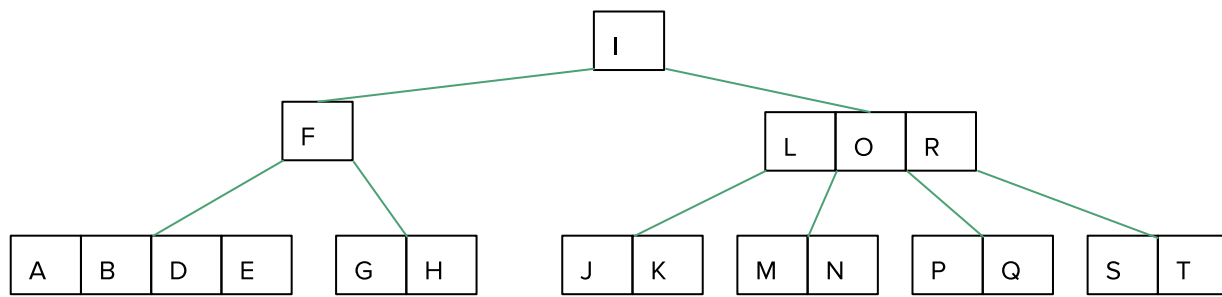
A B D E F G H I J K L In-Order ([MN]) O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L M N O In-Order ([PQ]) R In-Order ([ST])

A B D E F G H I J K L M N O P Q R In-Order ([ST])

ABDEFGHIJKLMNOPQRST

Pre-Order Traversal



Algorithm::Pre-Order

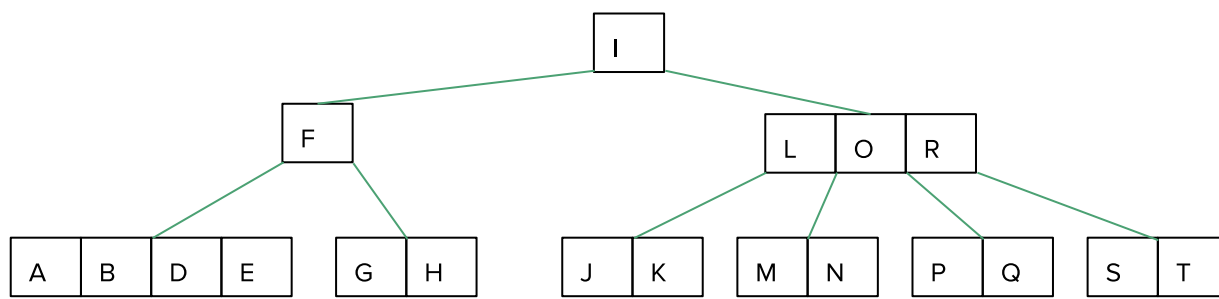
currentNodeEntries[$e_1 \dots e_n$] and currentNodeChildren[$c_1 \dots c_{n+1}$]

For $i=1:n$

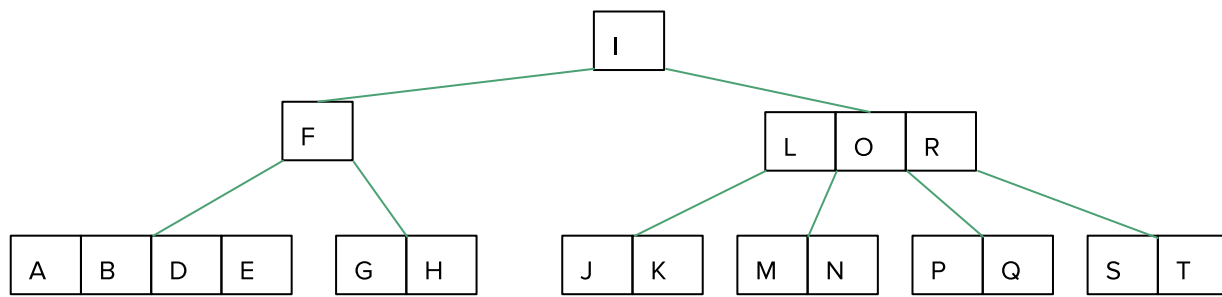
 Visit(currentNodeEntries(e_i))

for $i=1:n+1$

 Pre-Order(currentNodeChildren(c_i))

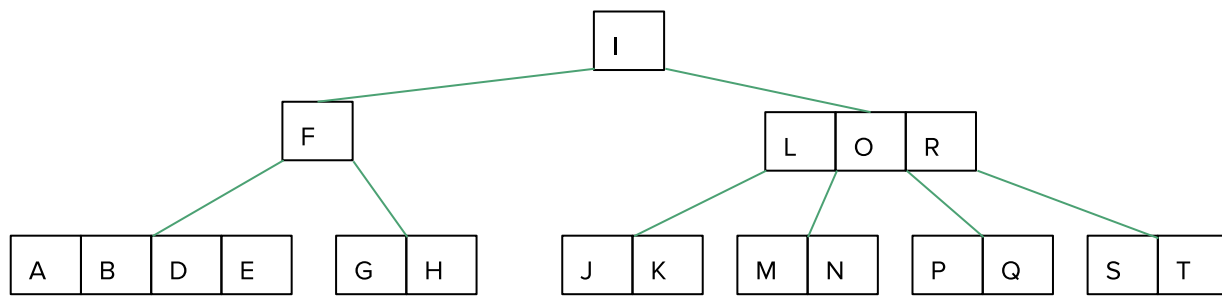


I Pre-Order ([F]) Pre-Order ([LOR])



I Pre-Order ([F]) Pre-Order ([LOR])

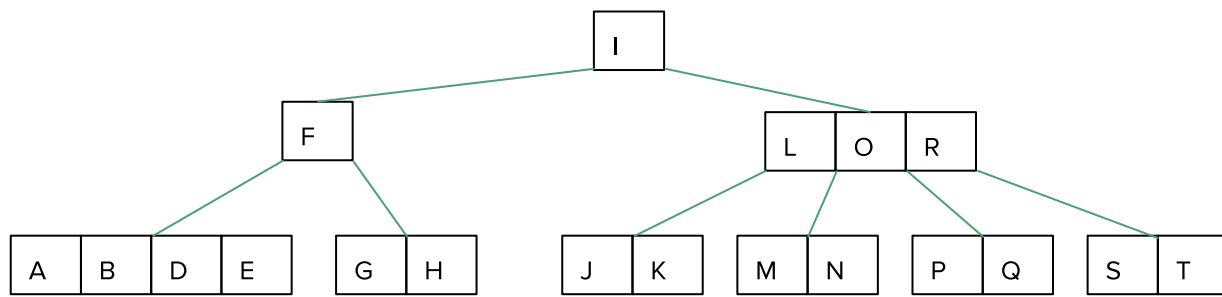
I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])



I Pre-Order ([F]) Pre-Order ([LOR])

I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])



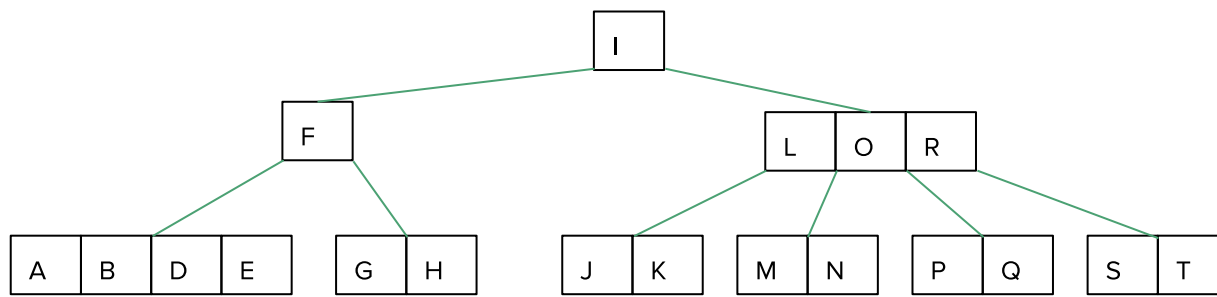
I Pre-Order ([F]) Pre-Order ([LOR])

I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E G H Pre-Order ([LOR])

T



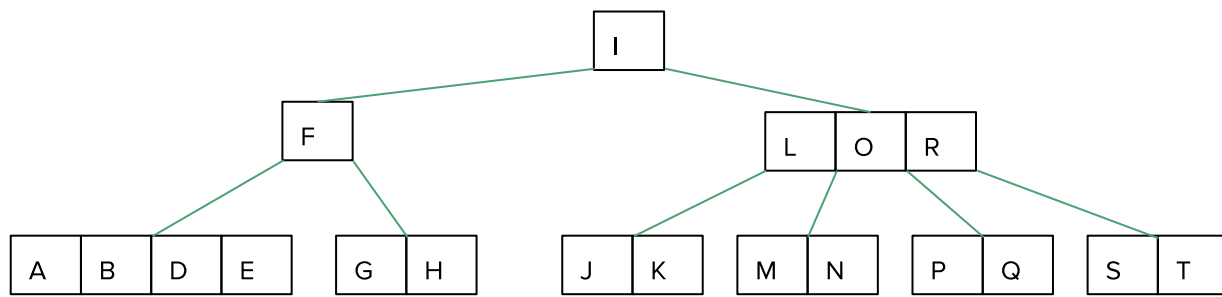
I Pre-Order ([F]) Pre-Order ([LOR])

I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E G H Pre-Order ([LOR])

I F A B D E G H L O R Pre-Order ([JK]) Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])



I Pre-Order ([F]) Pre-Order ([LOR])

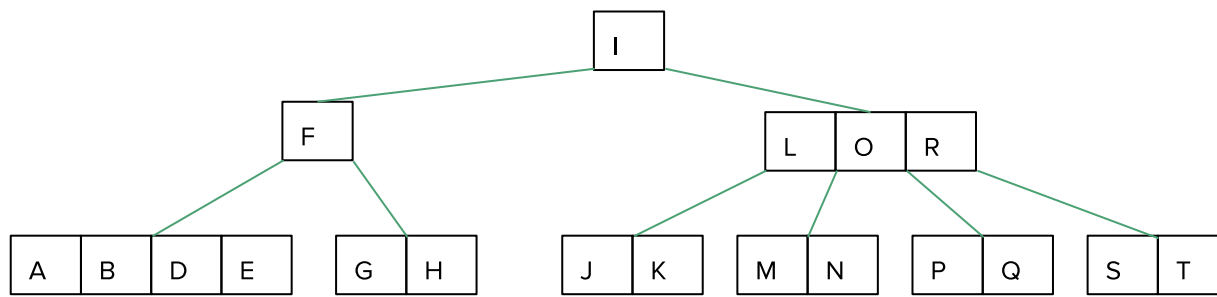
I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E G H Pre-Order ([LOR])

I F A B D E G H L O R Pre-Order ([JK]) Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])



I Pre-Order ([F]) Pre-Order ([LOR])

I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

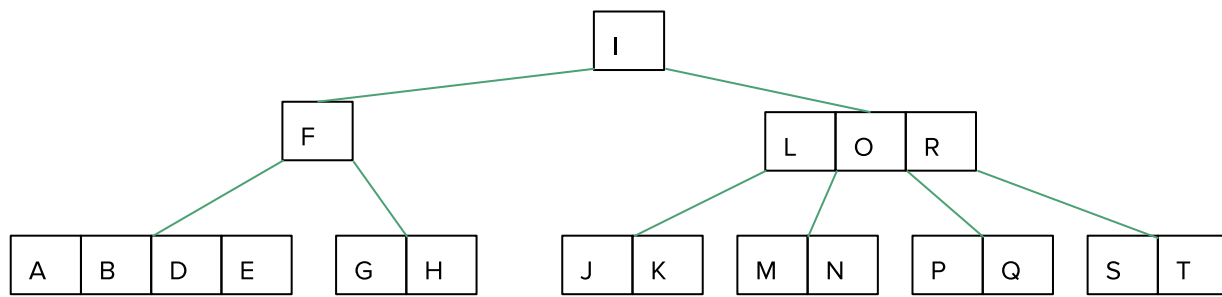
I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E G H Pre-Order ([LOR])

I F A B D E G H L O R Pre-Order ([JK]) Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K M N Pre-Order ([PQ]) Pre-Order ([ST])



I Pre-Order ([F]) Pre-Order ([LOR])

I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])

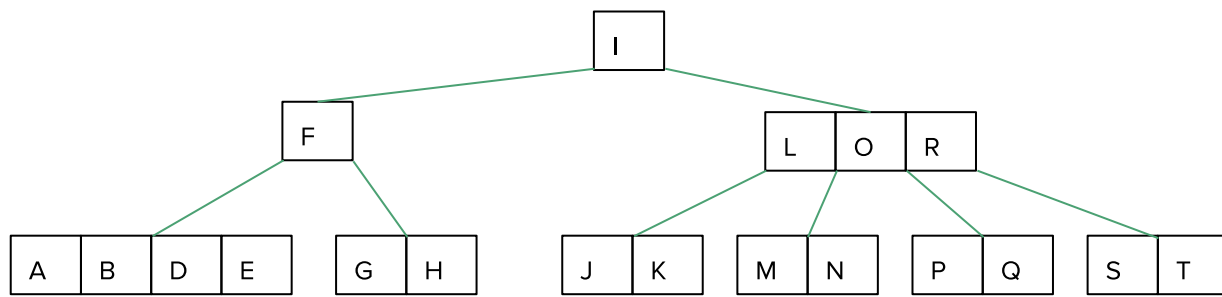
I F A B D E G H Pre-Order ([LOR])

I F A B D E G H L O R Pre-Order ([JK]) Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K M N Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K M N P Q Pre-Order ([ST])



I Pre-Order ([F]) Pre-Order ([LOR])

I F Pre-Order ([ABDE]) Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E Pre-Order ([GH]) Pre-Order ([LOR])

I F A B D E G H Pre-Order ([LOR])

I F A B D E G H L O R Pre-Order ([JK]) Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

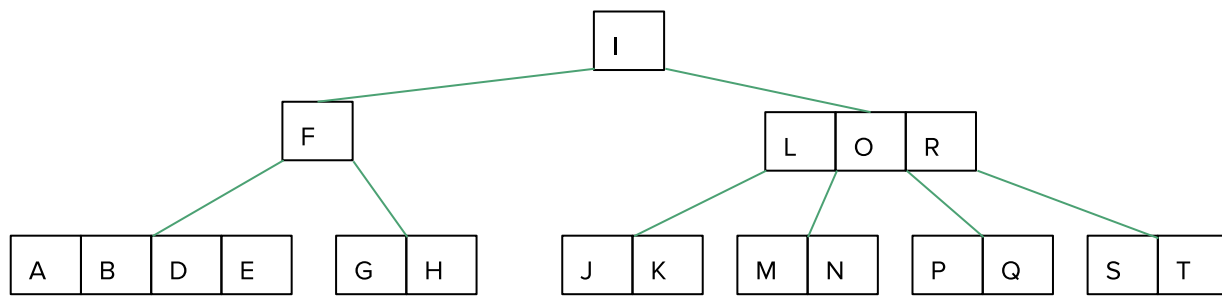
I F A B D E G H L O R J K Pre-Order ([MN]) Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K M N Pre-Order ([PQ]) Pre-Order ([ST])

I F A B D E G H L O R J K M N P Q Pre-Order ([ST])

I F A B D E G H L O R J K M N P Q S T

Post-Order Traversal



Algorithm::Post-Order

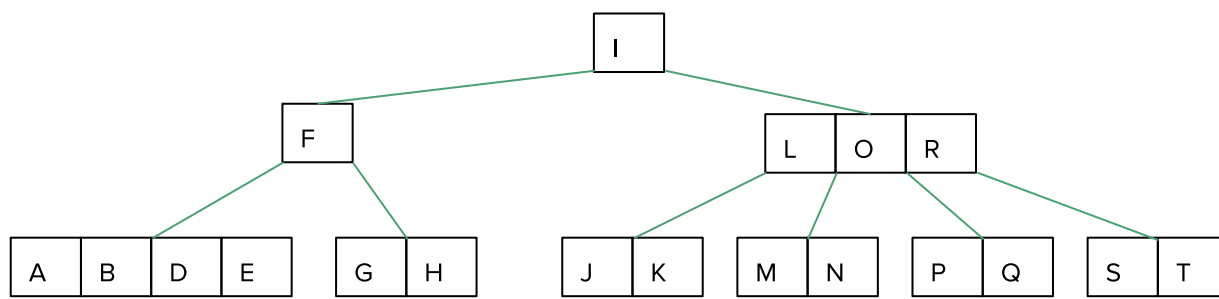
currentNodeEntries[$e_1 \dots e_n$] and currentNodeChildren[$c_1 \dots c_{n+1}$]

for $i=1:n+1$

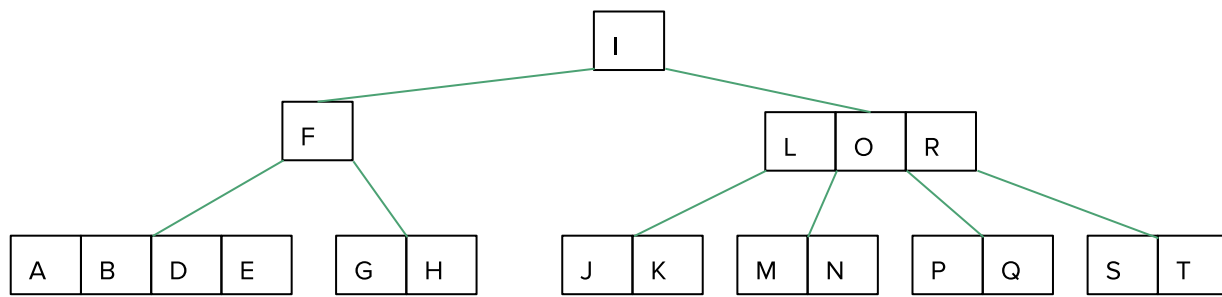
 Post-Order(currentNodeChildren(c_i))

For $i=1:n$

 Visit(currentNodeEntries(e_i))

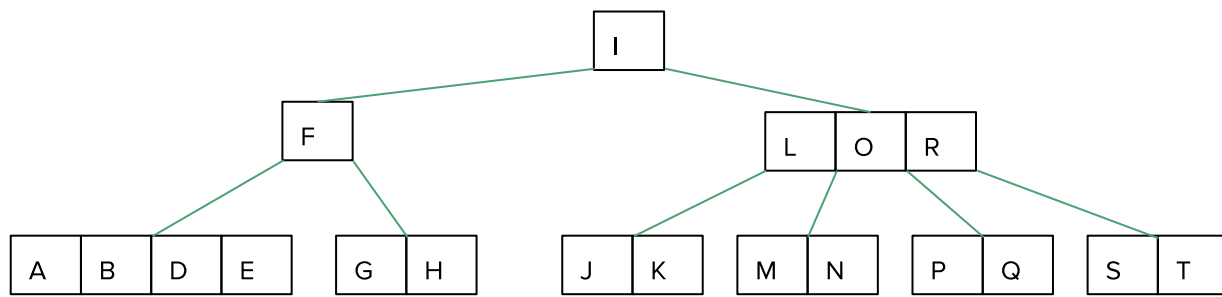


Post-Order ([F]) Post-Order ([LOR]) I



Post-Order ([F]) Post-Order ([LOR]) I

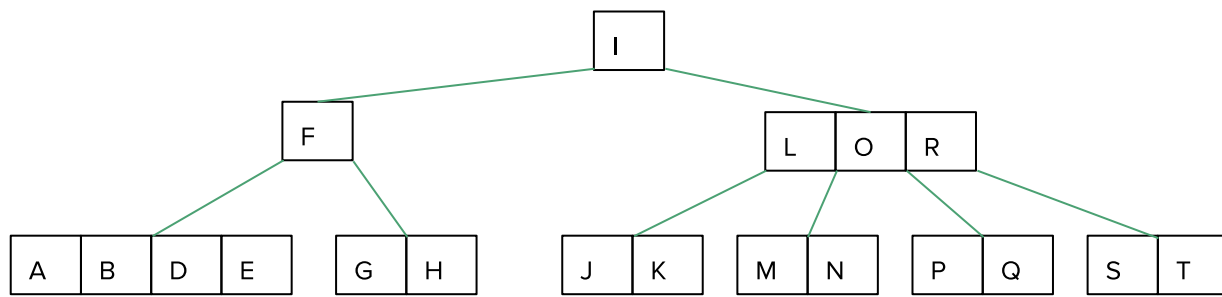
Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I



Post-Order ([F]) Post-Order ([LOR]) I

Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

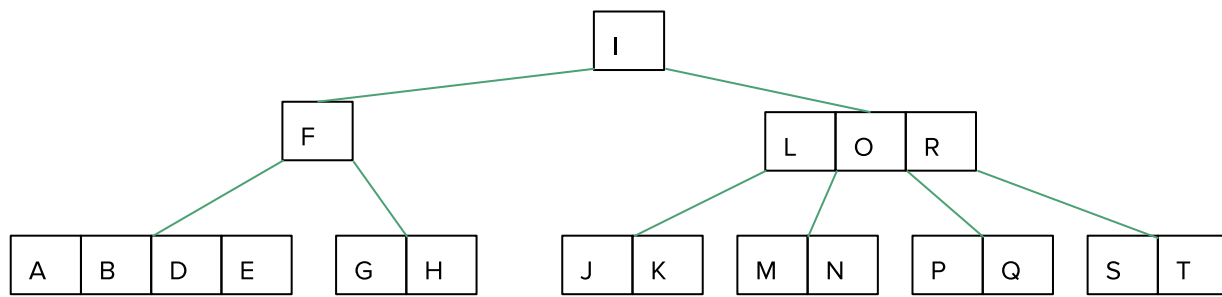


Post-Order ([F]) Post-Order ([LOR]) I

Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E G H F Post-Order ([LOR]) I



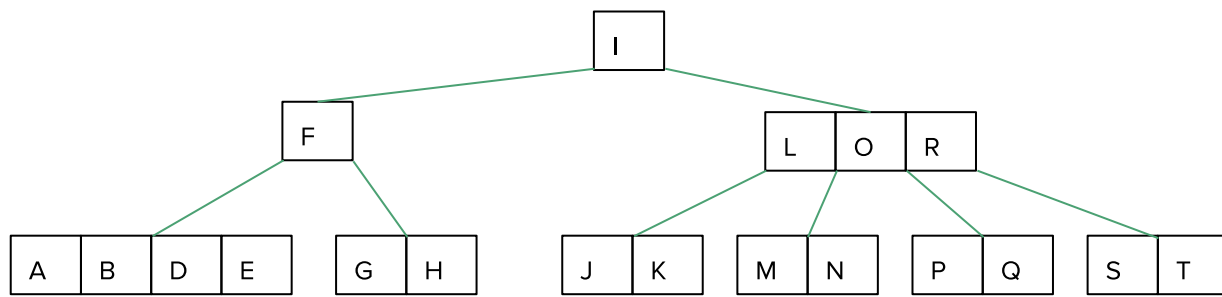
Post-Order ([F]) Post-Order ([LOR]) I

Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E G H F Post-Order ([LOR]) I

A B D E G H F Post-Order ([JK]) Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I



Post-Order ([F]) Post-Order ([LOR]) I

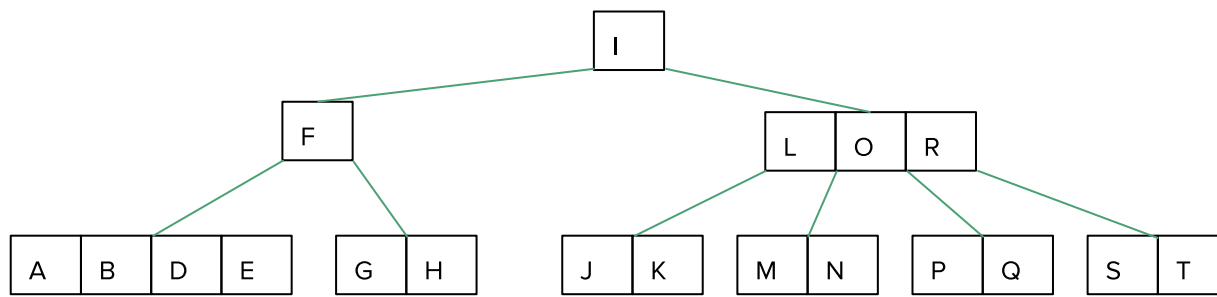
Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E G H F Post-Order ([LOR]) I

A B D E G H F Post-Order ([JK]) Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I



Post-Order ([F]) Post-Order ([LOR]) I

Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

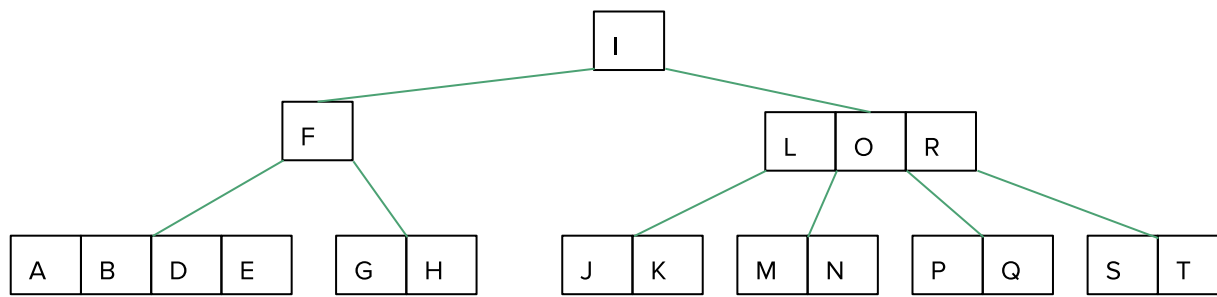
A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E G H F Post-Order ([LOR]) I

A B D E G H F Post-Order ([JK]) Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K M N Post-Order ([PQ]) Post-Order ([ST]) L O R I



Post-Order ([F]) Post-Order ([LOR]) I

Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

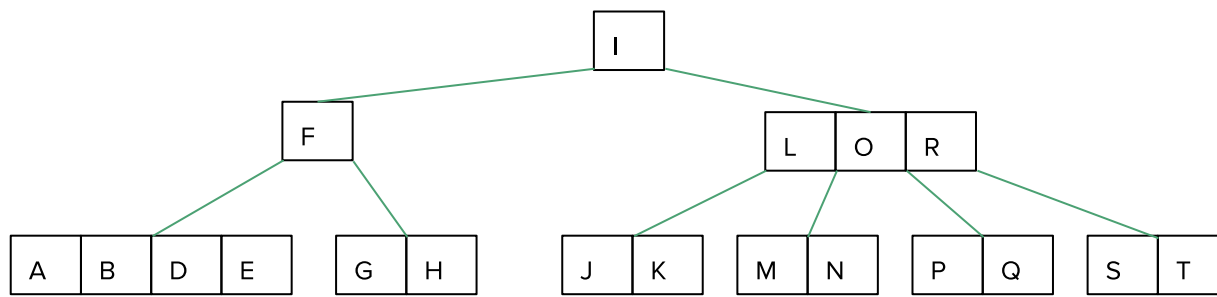
A B D E G H F Post-Order ([LOR]) I

A B D E G H F Post-Order ([JK]) Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K M N Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K M N P Q Post-Order ([ST]) L O R I



Post-Order ([F]) Post-Order ([LOR]) I

Post-Order ([ABDE]) Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E Post-Order ([GH]) F Post-Order ([LOR]) I

A B D E G H F Post-Order ([LOR]) I

A B D E G H F Post-Order ([JK]) Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K Post-Order ([MN]) Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K M N Post-Order ([PQ]) Post-Order ([ST]) L O R I

A B D E G H F J K M N P Q Post-Order ([ST]) L O R I

ABDEGHFJKMNPQSTLORI