

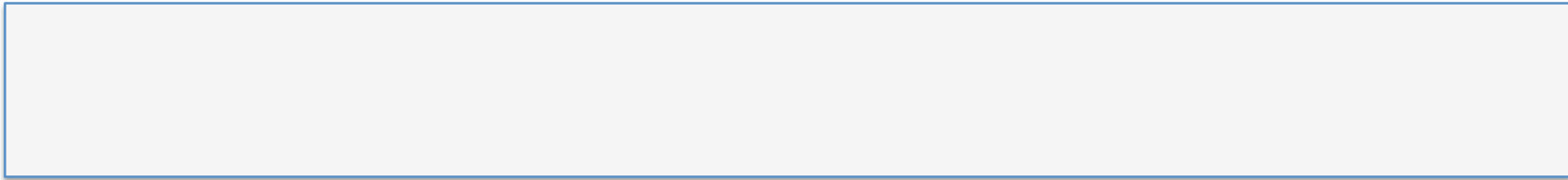
B-Trees

Problem: Data base too big to fit memory
Disk reads are slow

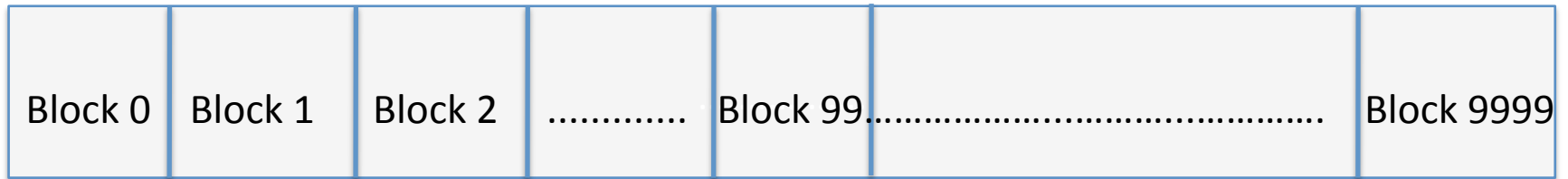
Example: 1,000,000 records on disk
Binary search might take
20 disk reads

Disk reads are done in blocks

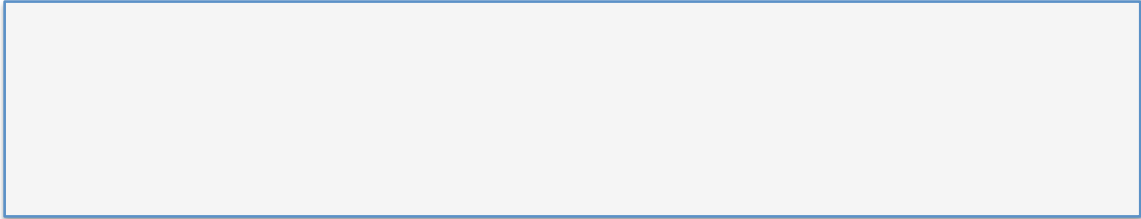
Example: One block read can retrieve
100 records



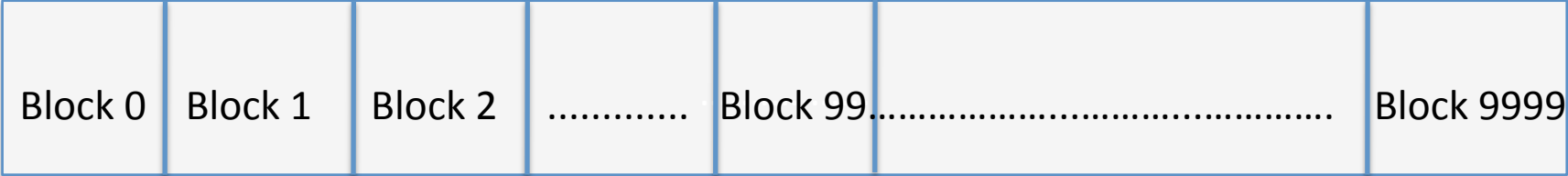
1,000,000 Records



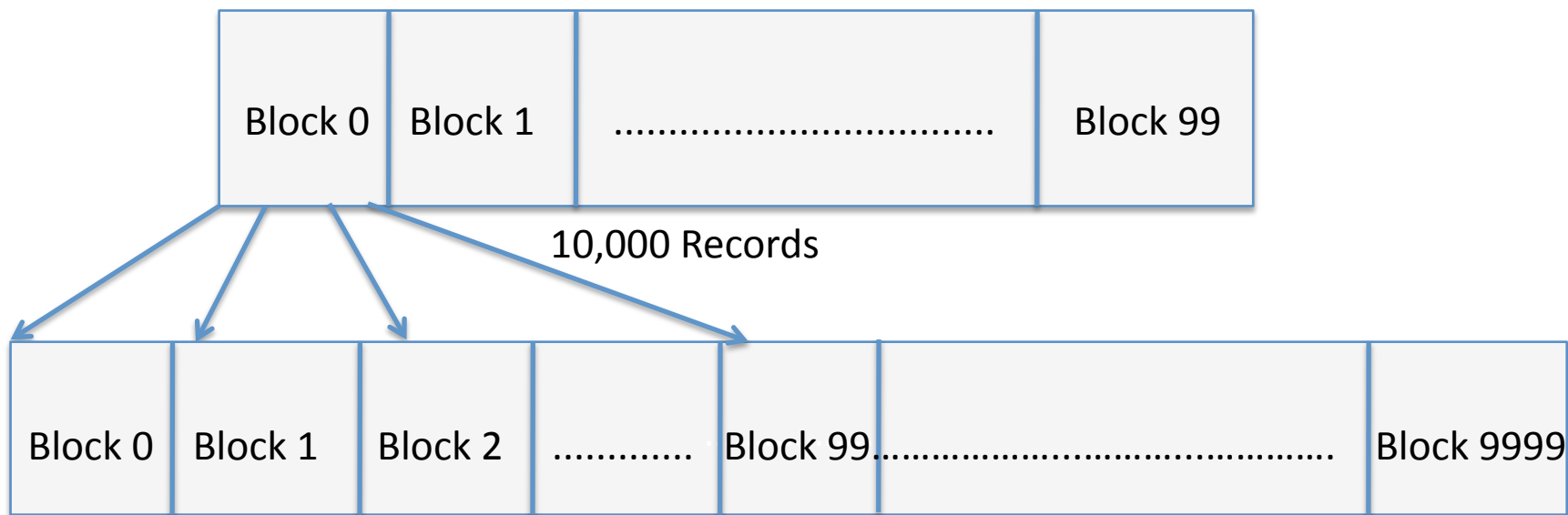
1,000,000 Records



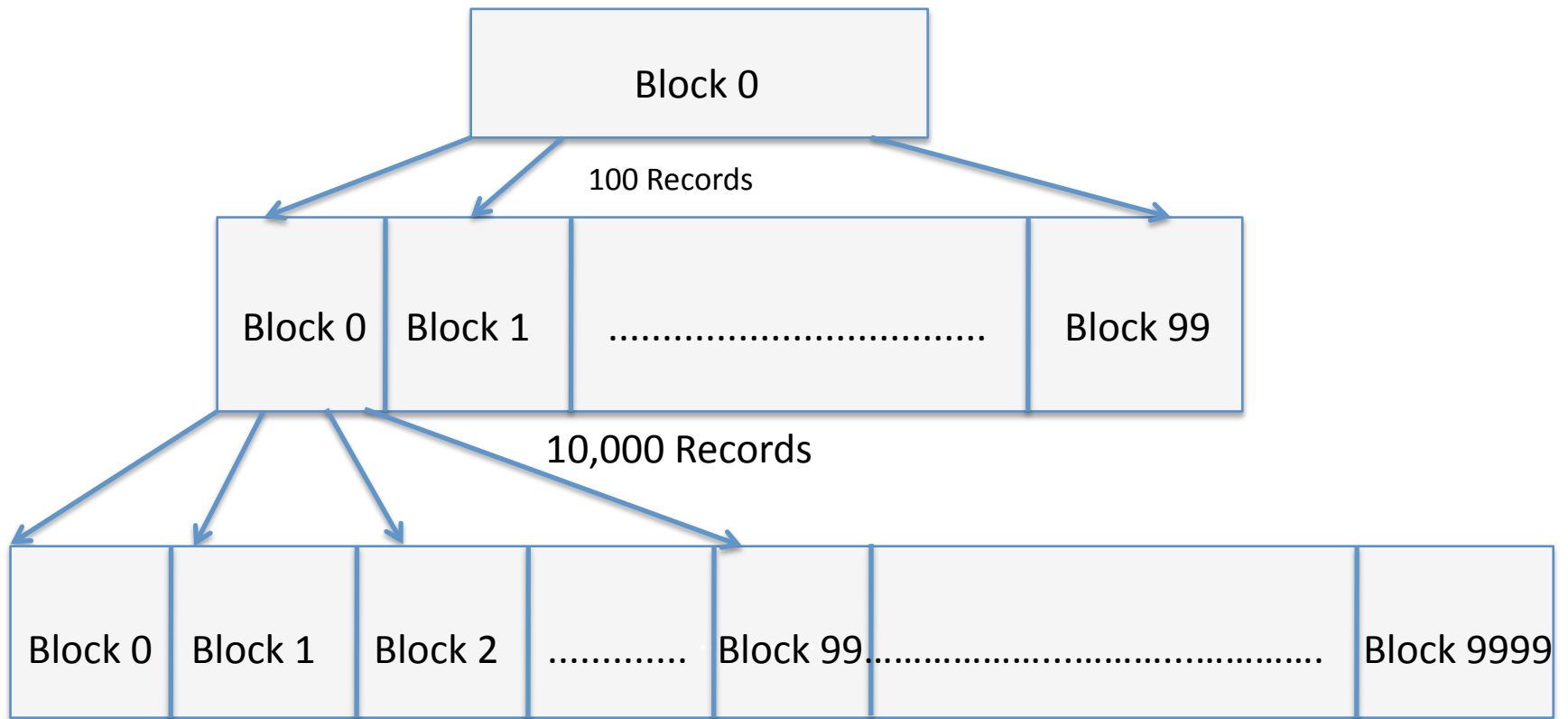
10,000 Records



1,000,000 Records



1,000,000 Records



1,000,000 Records

DEF: A B-Tree of order m is an m -way tree such that

1. All leaf nodes are at the same level.
2. All non-leaf nodes (except the root) have at most m and at least $m/2$ children.
3. The number of keys is one less than the number of children for non-leaf nodes and at most $m-1$ and at least $m/2$ for leaf nodes.
4. The root may have as few as 2 children unless the tree is the root alone.

Example for $m = 5$

DEF: A B-Tree of order 5 is an 5-way tree such that

1. All leaf nodes are at the same level.
2. All non-leaf nodes (except the root) have at most 5 and at least 2 children.
3. The number of keys is one less than the number of children for non-leaf nodes and at most 4 and at least 2 for leaf nodes.
4. The root may have as few as 2 children unless the tree is the root alone.

Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P

Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



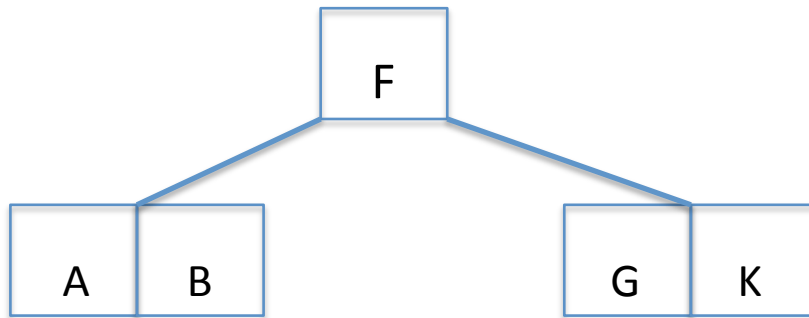
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P

A	B	F	G	K
---	---	---	---	---

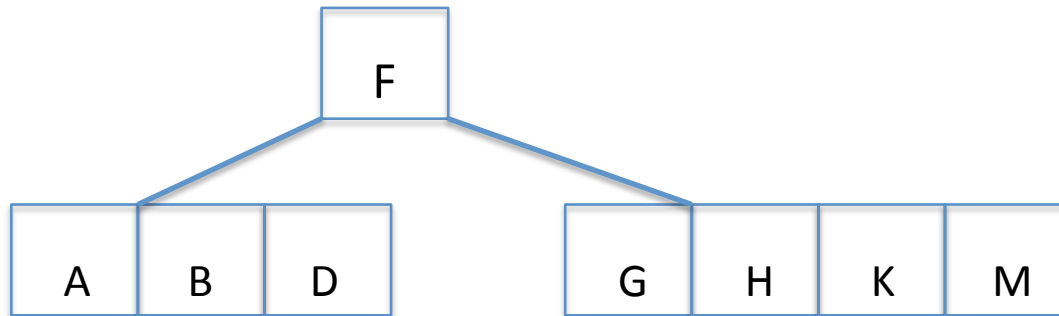
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



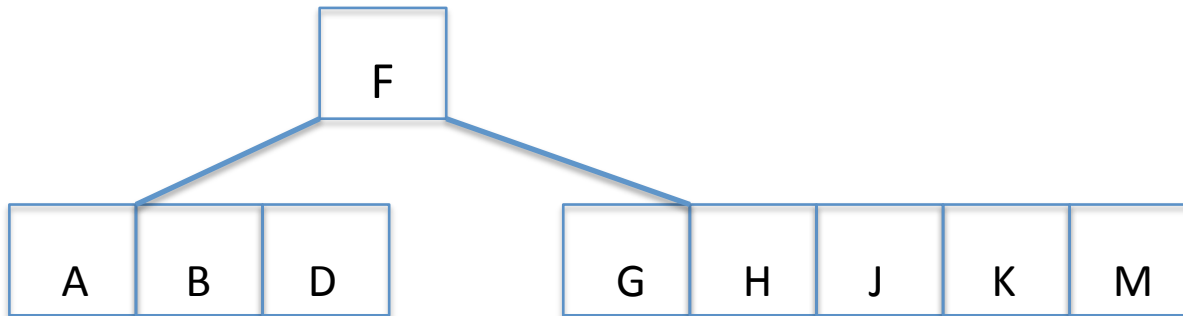
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



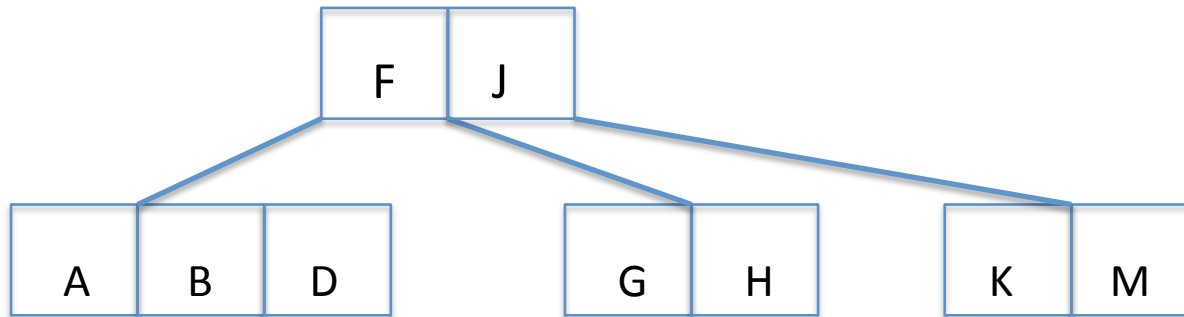
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



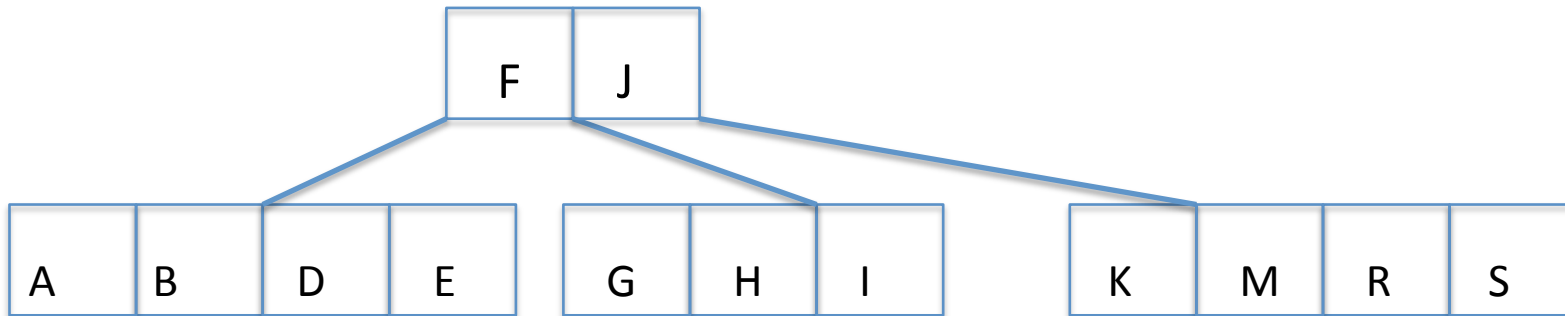
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



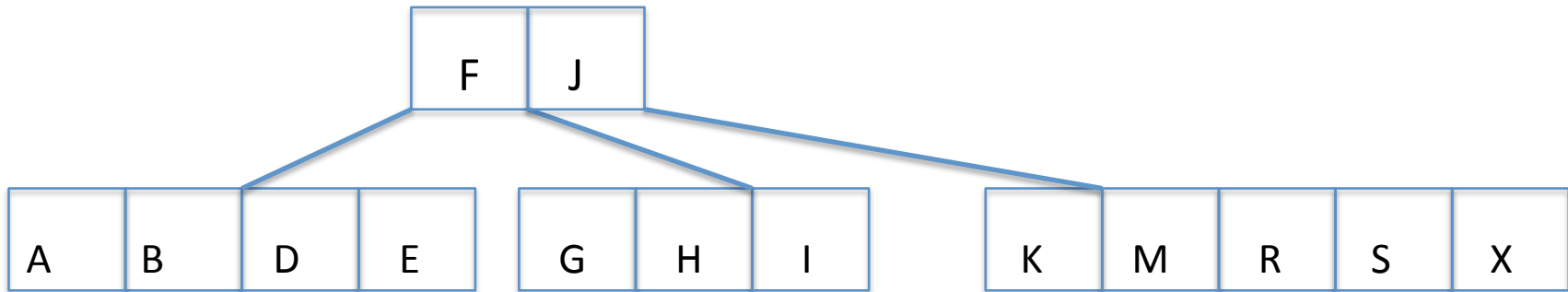
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



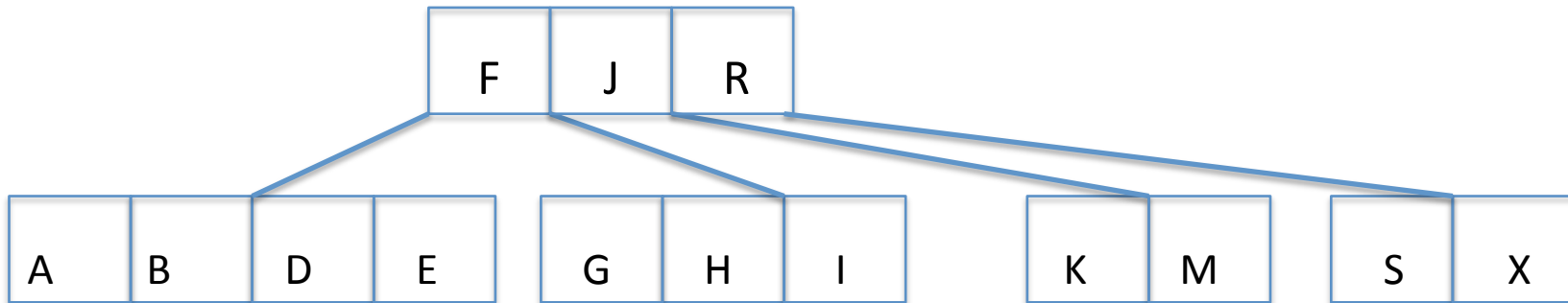
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



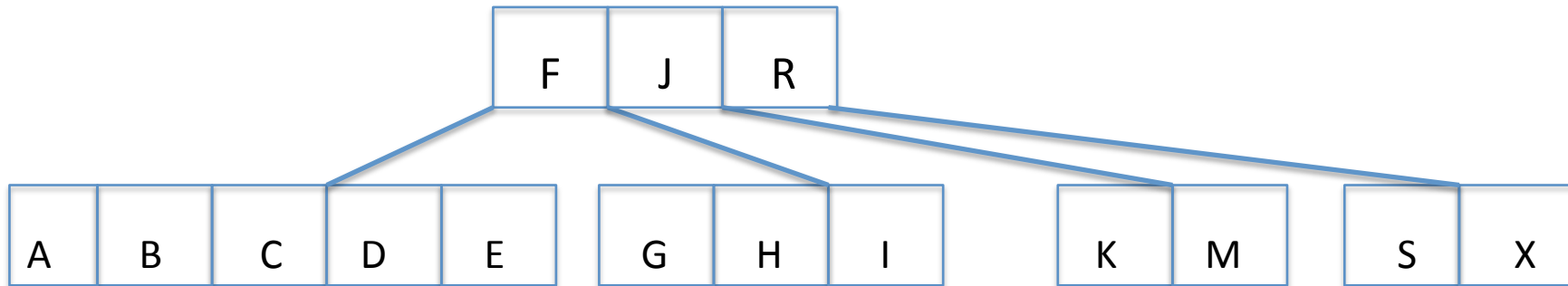
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



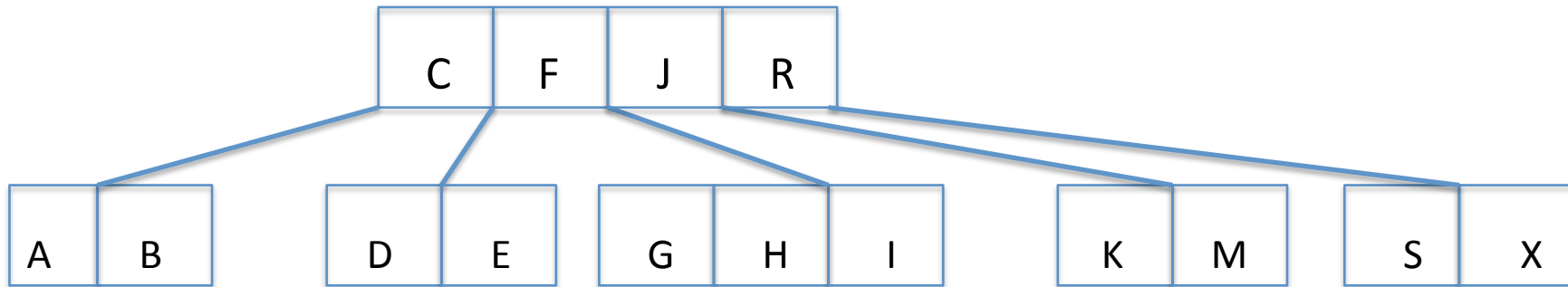
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



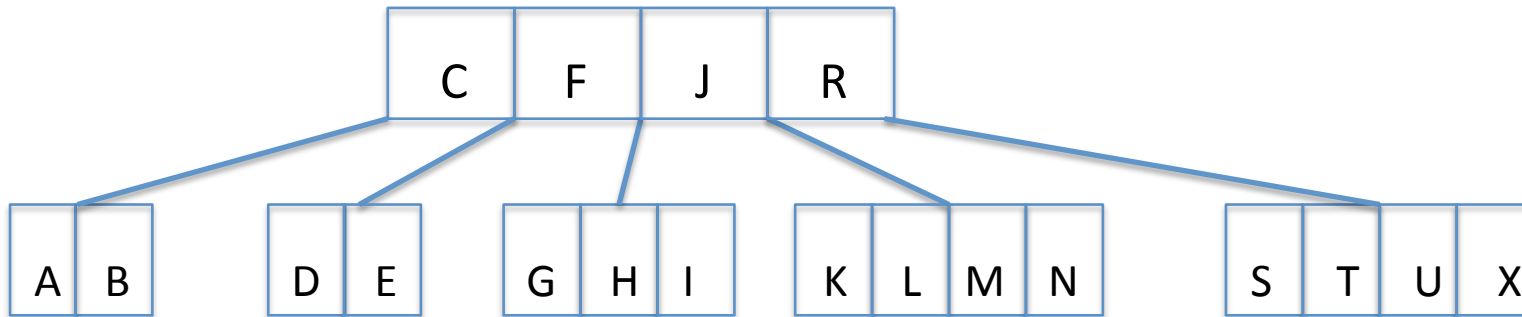
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



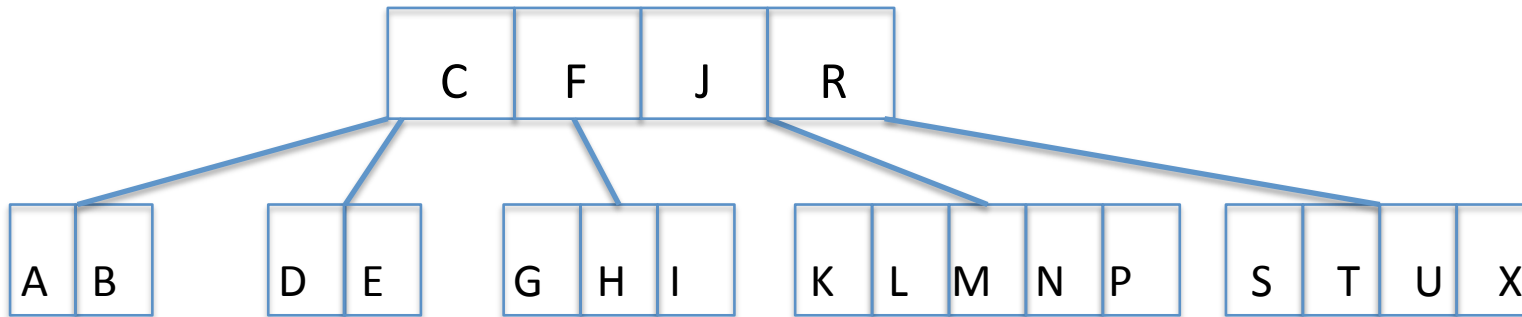
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



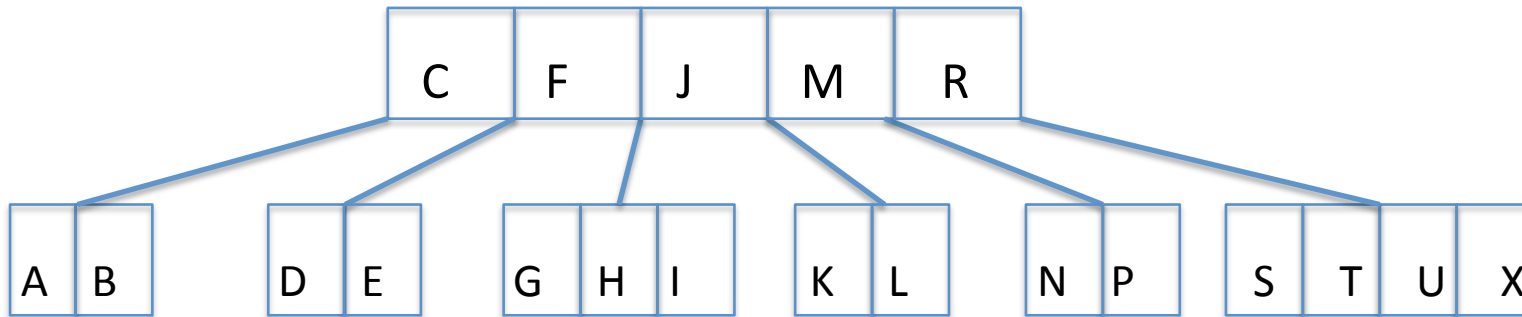
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



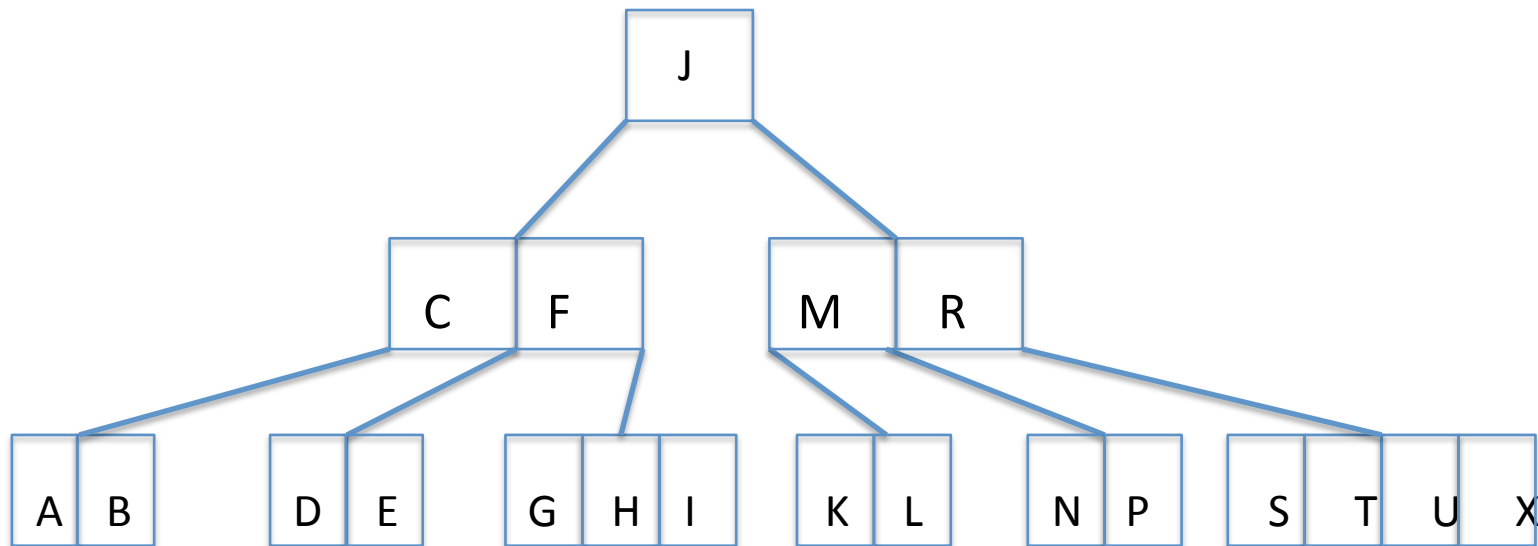
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



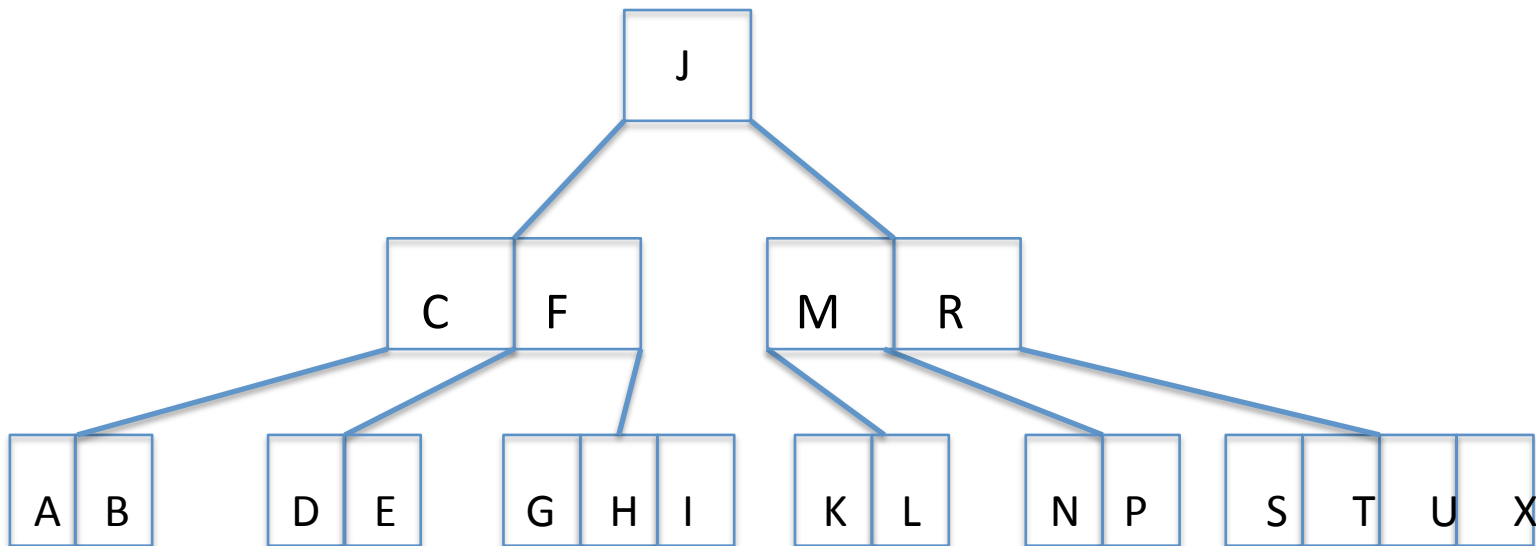
Creating a B-tree of order 5

A G F B K D H M J E S I R X C L N T U P



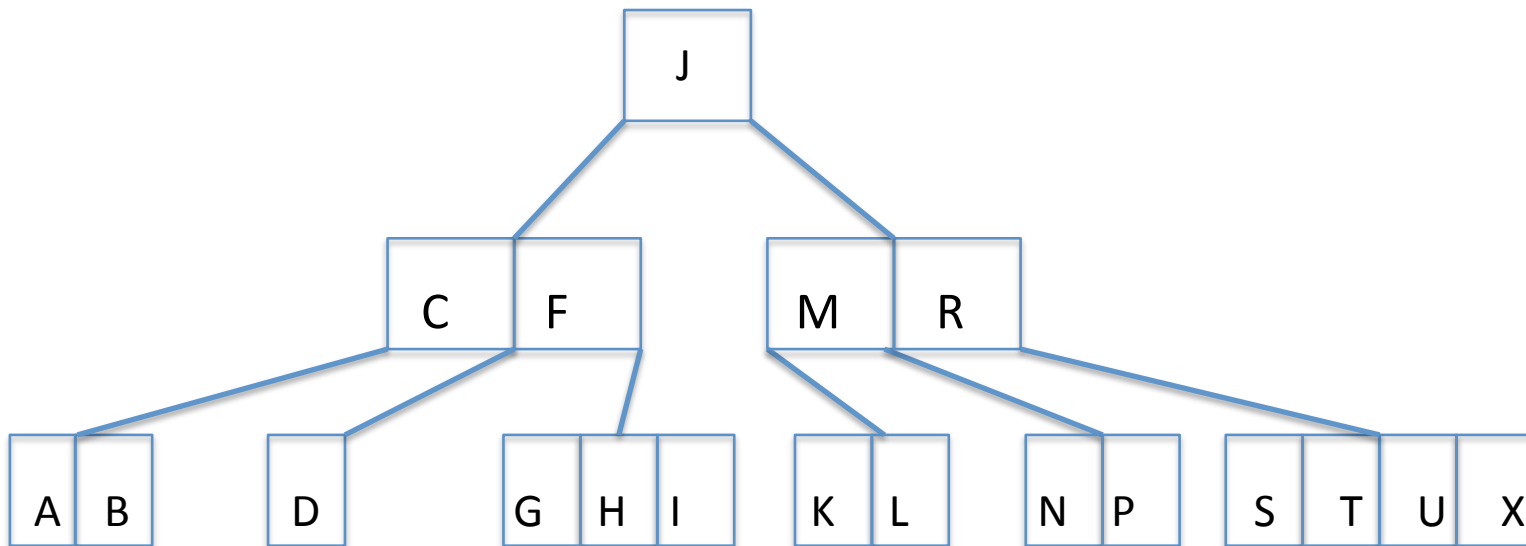
Deleting Nodes

- Delete E from leaf node



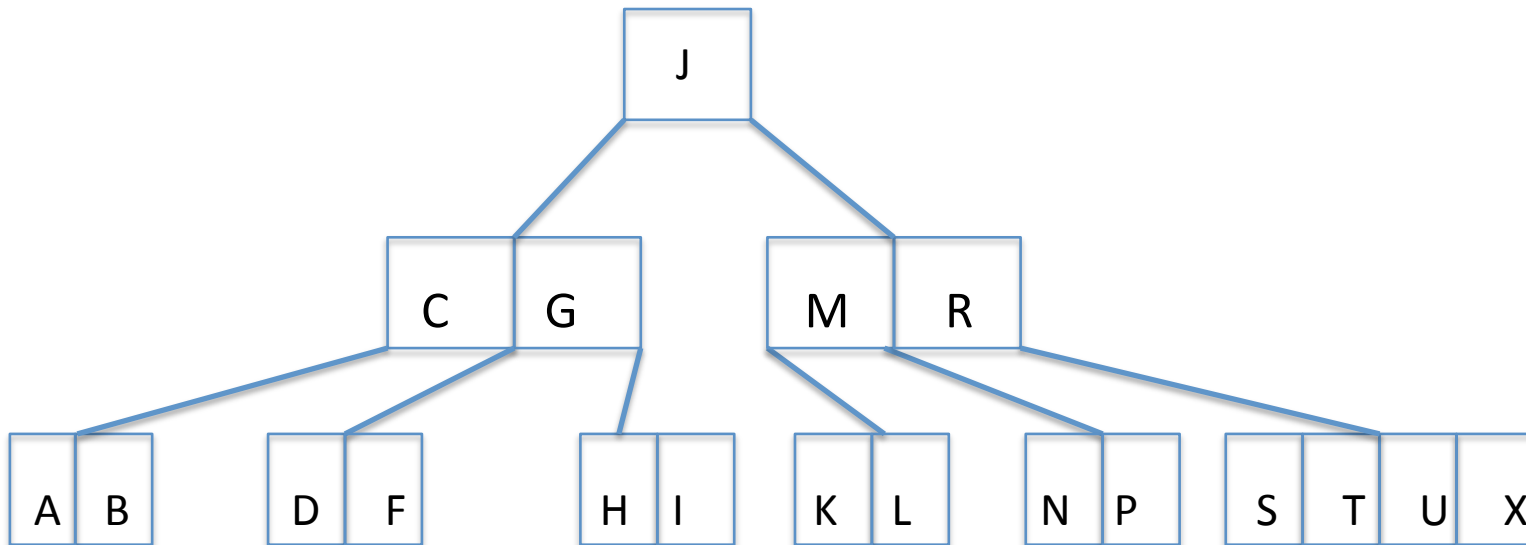
Deleting Nodes

- Delete E



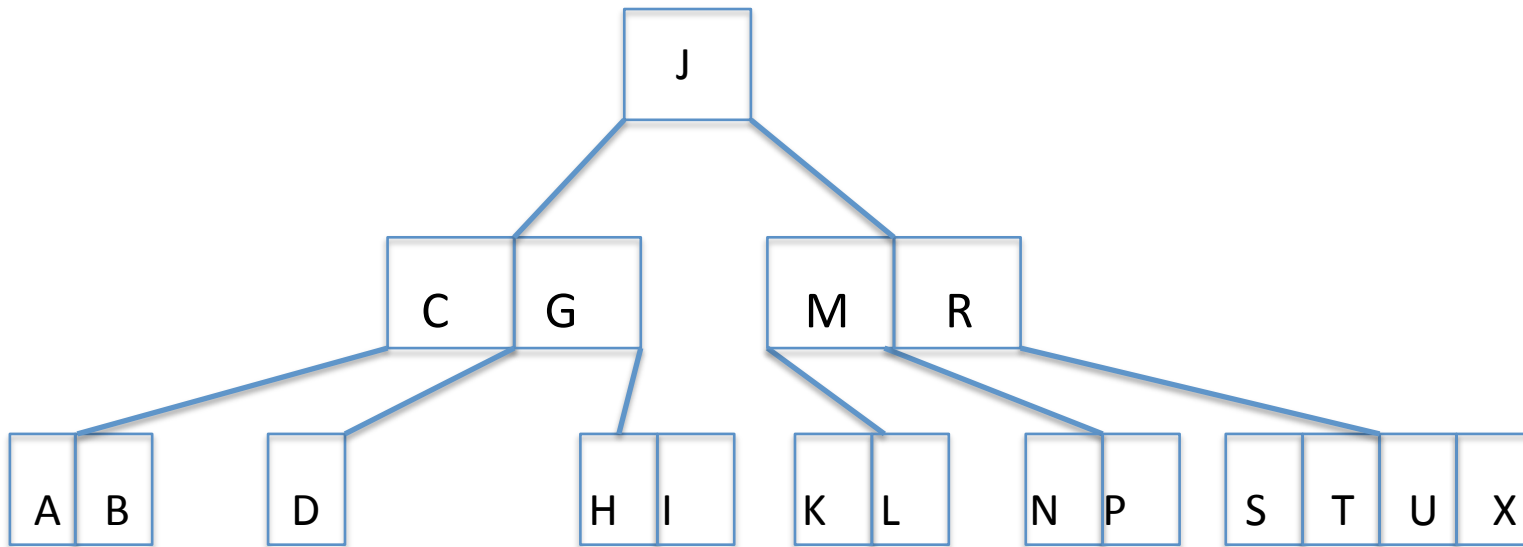
Deleting Nodes

- Borrow from a neighbor



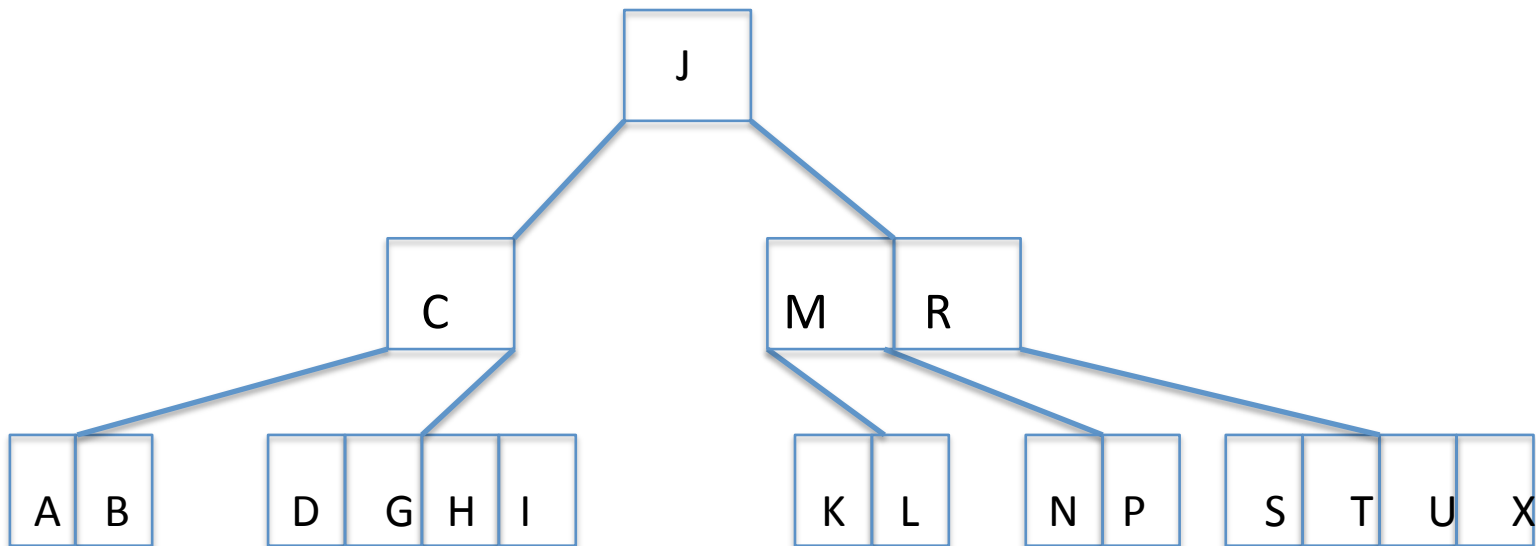
Deleting Nodes

- Delete F --- but can't borrow from a neighbor



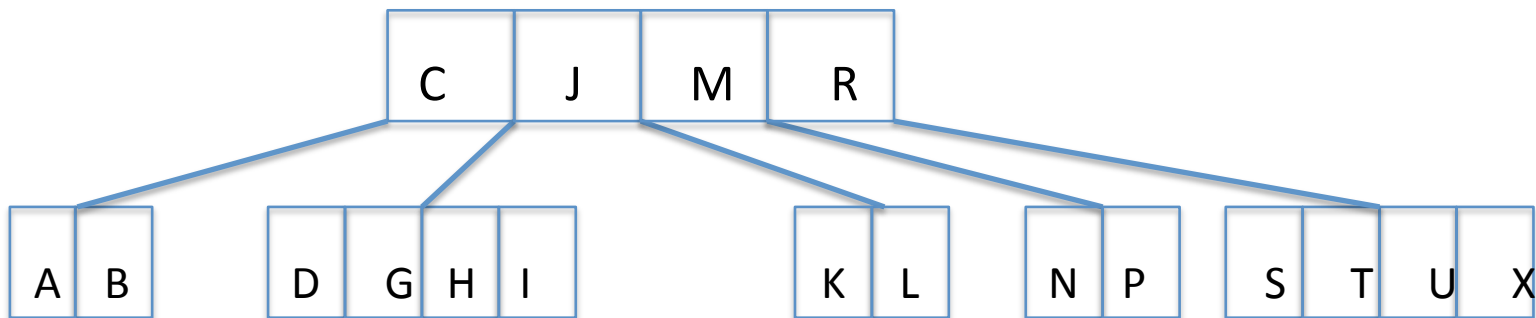
Deleting Nodes

Combine and push the problem up one level



Deleting Nodes

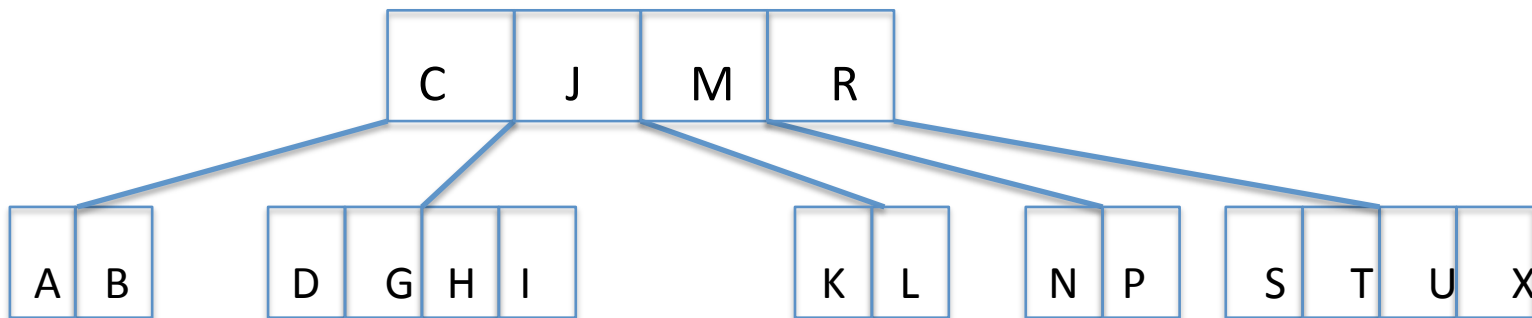
Can't borrow so combine



Deleting Nodes

Delete M from non-leaf node

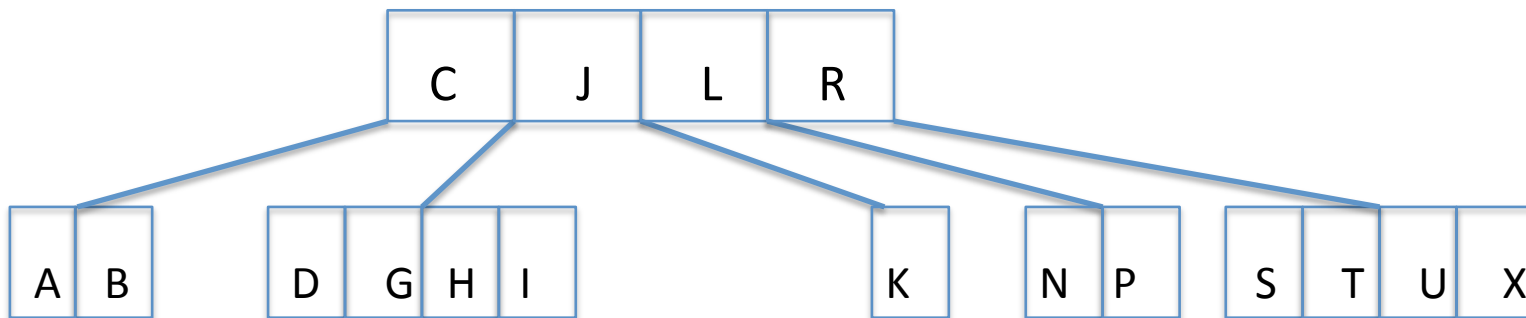
Note: immediate predecessor in non-leaf
Is always in a leaf.



Deleting Nodes

Delete M from non-leaf node

Overwrite M with immediate predecessor



Deleting Nodes

Borrow from a neighbor

