

CS 4/56101	Kent State University Dept. of Math & Computer Science <u>LECT-12</u>
Design and Analysis of Algorithms	

Multiway Trees

Tree

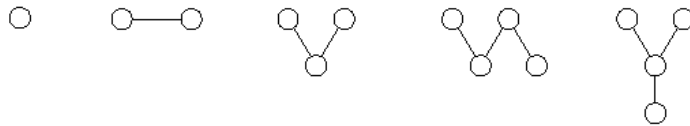
- A *(free) tree* is any set of points (called **vertices**) and any set of pairs of distinct vertices (called **edges** or **branches**) such that (1) there is a sequence of edges (a **path**) from any vertex to any other, and (2) there are no **circuits**, that is, no paths starting from a vertex and returning to the same vertex.
- A **rooted tree** is a tree in which one vertex, called the **root**, is distinguished.
- An **ordered tree** is a rooted tree in which the children of each vertex are assigned an order.
- A **forest** is a set of trees. We usually assume that all trees in a forest are rooted.
- An **orchard** (also called an **ordered forest**) is an ordered set of ordered trees.



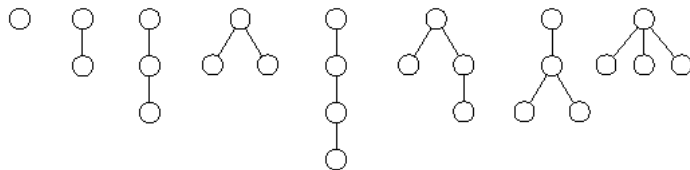
DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-3
ALG00S, javed@kent.edu
Javed I. Khan@1999

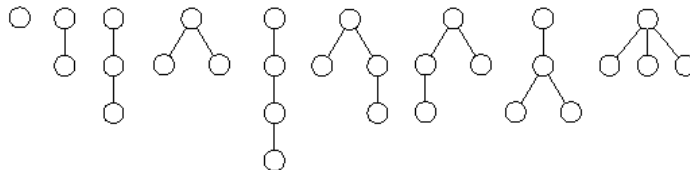
Examples



Free trees with 4 or fewer vertices
(Arrangement of vertices is irrelevant.)



Rooted trees with 4 or fewer vertices



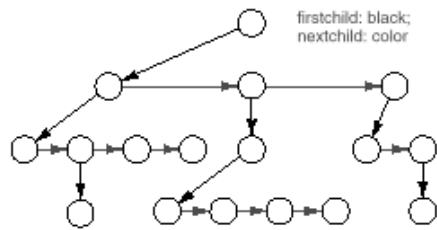
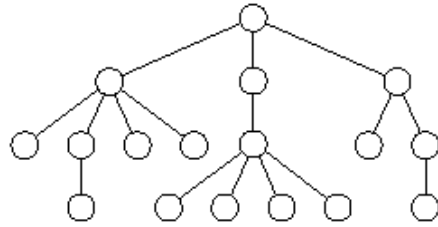
Ordered trees with 4 or fewer vertices



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-4
ALG00S, javed@kent.edu
Javed I. Khan@1999

Implementation

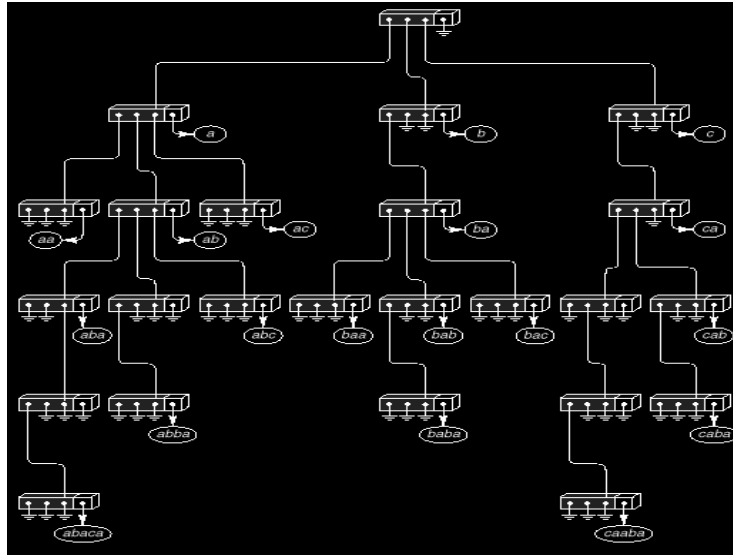


DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-5
ALG00S, javed@kent.edu
Javed I. Khan © 1999

Tries

Tries



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-7
ALG00S, javed@kent.edu
Javed I. Khan@1999

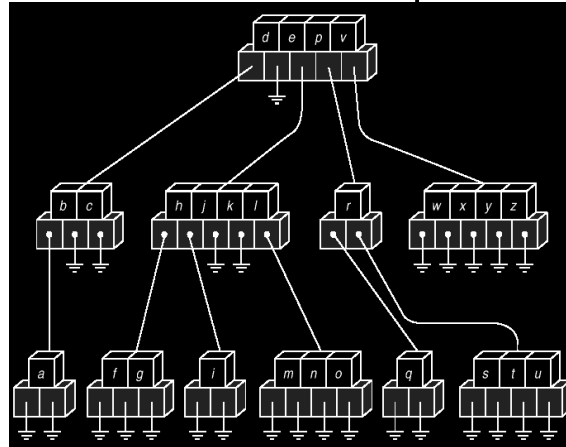
B-Trees

B-Trees



DESIGN &
ANALYSIS OF
ALGORITHM

- Idea: In external search disk access is expensive. How can we store more than 2-way information in a block?
- Solution: construct multiway search tree.



LECT-12, S-9
ALG00S, javed@kent.edu
Javed I. Khan@1999

B-Trees

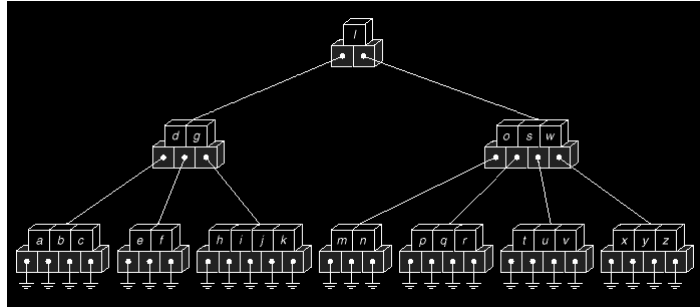


DESIGN &
ANALYSIS OF
ALGORITHM

- A *B-tree of order m* is an m -way search tree in which
 - All leaves are on the same level.
 - All internal nodes except the root have at most m non-empty children, and at least $\lceil m/2 \rceil$ nonempty children.
 - The number of keys in each internal node is one less than the number of its children, and these keys partition the keys in the children in the fashion of a search tree.
 - The root has at most m children, or as few as 2 if it is not a leaf, or none if the tree consists of the root alone.

LECT-12, S-10
ALG00S, javed@kent.edu
Javed I. Khan@1999

Example (m=4)



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-11
ALG00S, javed@kent.edu
Javed I. Khan@1999

Insertion

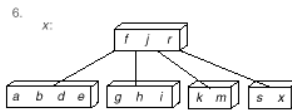
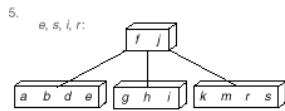
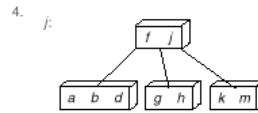
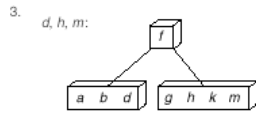
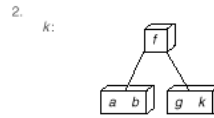
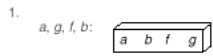
- B-trees grow at the root, not the leaves.
- Find the leaf where the new key belongs and insert it.
- If the leaf now has too many keys, split it into two nodes on the same level, but do not put the median key into either new node.
- Move up one level, insert the median key in this parent node, and repeat the splitting process if necessary.
- If the root node splits, then the resulting median key goes into a new root and the tree grows in height.



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-12
ALG00S, javed@kent.edu
Javed I. Khan@1999

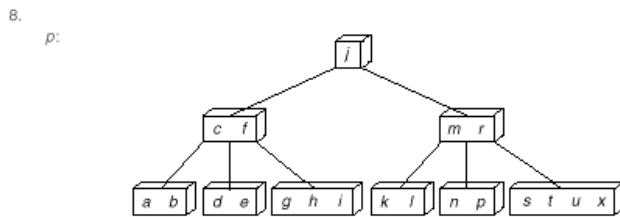
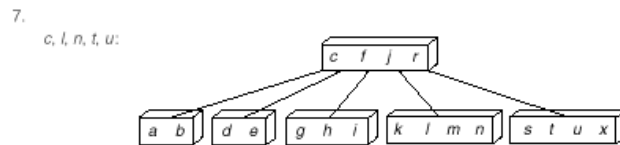
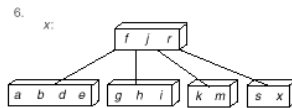
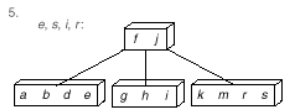
Insertion Example



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-13
ALG00S, javed@kent.edu
Javed I. Khan@1999

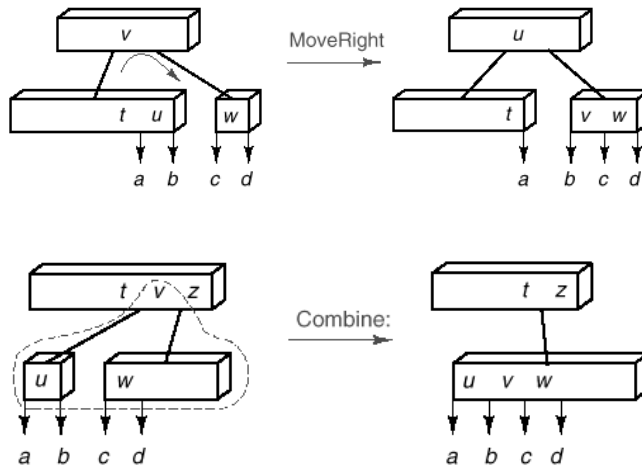
Insertion Example



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-14
ALG00S, javed@kent.edu
Javed I. Khan@1999

Deletion

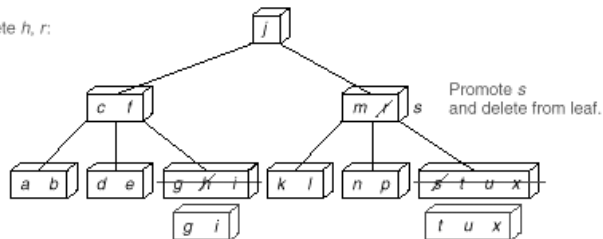


DESIGN &
ANALYSIS OF
ALGORITHM

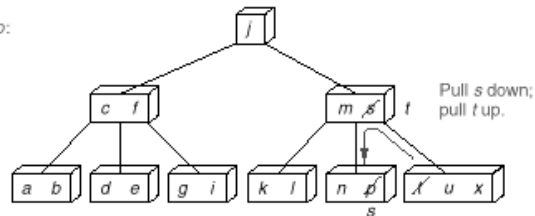
LECT-12, S-15
ALG00S, javed@kent.edu
Javed I. Khan@1999

Deletion Example

1. Delete h, r :



2. Delete p :

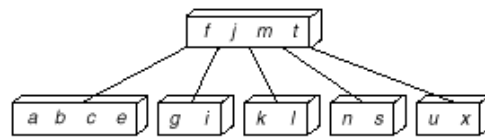
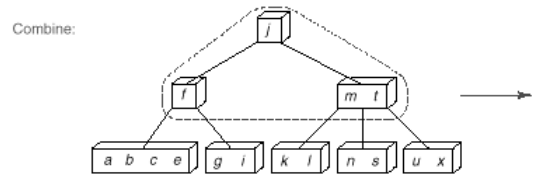
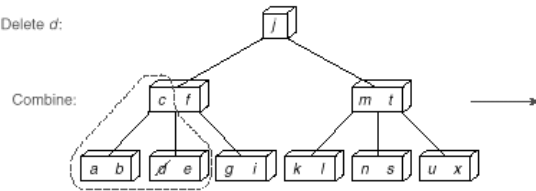


DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-16
ALG00S, javed@kent.edu
Javed I. Khan@1999

Deletion Example (cont..)

3. Delete *d*:



DESIGN &
ANALYSIS OF
ALGORITHM

LECT-12, S-17
ALG00S, javed@kent.edu
Javed I. Khan@1999