


Prev topic

Next topic

Next page

Prev page

The Lecture Contains:

 X-tree

- Insertion and splitting
- Overlap-minimal split
- Deletion, searching and performance

Prev topic

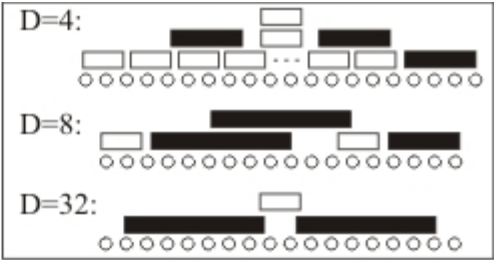
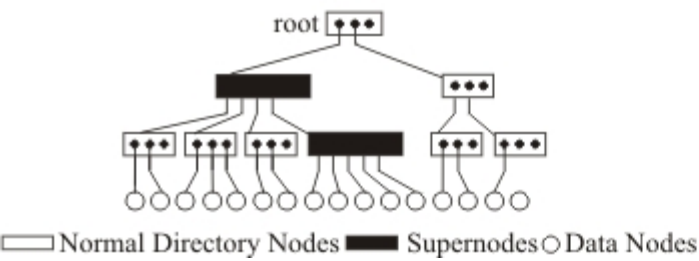
Next topic

Prev page

Next page

X-tree

- Node splits are avoided if they are bad, i.e., overlaps are large
- A **supernode** is created that spans two (sequential) disk pages
- Supernodes may be extended and can span arbitrary number of disk pages
- Number and size of supernodes increase with dimensions



Module 5: Disk-based Index Structures

Lecture 23: X-tree

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Insertion and splitting

- Two policies of splitting
 - Topological split
 - Overlap-minimal split
- Split only if overlap below threshold
- Threshold determined by access time parameters

Module 5: Disk-based Index Structures

Lecture 23: X-tree

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Insertion and splitting

- Two policies of splitting
 - Topological split
 - Overlap-minimal split
- Split only if overlap below threshold
- Threshold determined by access time parameters
- Modifying and writing supernodes may require multiple disk accesses
 - Assumes availability of contiguous pages on disk
- Topological split is similar to that of R*-tree
- Overlap-minimal split

Module 5: Disk-based Index Structures

Lecture 23: X-tree

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Overlap-minimal split

- **Split tree** is maintained to ascertain which dimensions have been split
 - Internal node encodes **split history**
- Dimension at the root of the split is the split axis
 - For point data, overlap-free split is possible

Module 5: Disk-based Index Structures

Lecture 23: X-tree

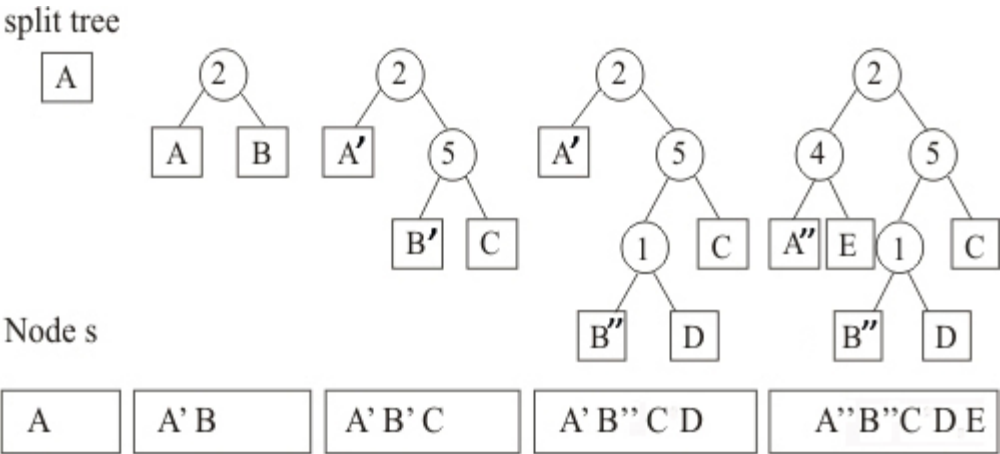
[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Overlap-minimal split

- Split tree is maintained to ascertain which dimensions have been split
 - Internal node encodes split history
- Dimension at the root of the split is the split axis
 - For point data, overlap-free split is possible
- May lead to underflow, and split may be rejected

Overlap-minimal split

- Split tree is maintained to ascertain which dimensions have been split
 - Internal node encodes split history
- Dimension at the root of the split is the split axis
 - For point data, overlap-free split is possible
- May lead to underflow, and split may be rejected
- Requires extra storage of split history at each node



Prev topic

Next topic

Prev page

Next page

- Deletion, searching and performance
- Underflow of supernodes may create normal nodes

Module 5: Disk-based Index Structures

Lecture 23: X-tree

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Deletion, searching and performance

- Underflow of supernodes may create normal nodes
- When both pages need to be accessed, **X**-tree reduces the cost by putting them sequentially rather than random I/O

Module 5: Disk-based Index Structures

Lecture 23: X-tree

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Deletion, searching and performance

- Underflow of supernodes may create normal nodes
- When both pages need to be accessed, **X**-tree reduces the cost by putting them sequentially rather than random I/O
- Factor of speed-up over R*-tree is high for higher dimensions
- Combination of efficient behavior of
 - R*-trees in low dimensions
 - Sequential scan in high dimensions

Module 5: Disk-based Index Structures

Lecture 23: X-tree

[Prev topic](#)[Next topic](#)[Prev page](#)[Next page](#)

Deletion, searching and performance

- Underflow of supernodes may create normal nodes
- When both pages need to be accessed, **X**-tree reduces the cost by putting them sequentially rather than random I/O
- Factor of speed-up over R*-tree is high for higher dimensions
- Combination of efficient behavior of
 - R*-trees in low dimensions
 - Sequential scan in high dimensions
- Still does not scale well for very high dimensions in comparison to sequential scan