Module 5: Disk-based Index Structures

Lecture 18: General Framework

The Lecture Contains:

- General framework for hierarchical object-based index structures
- Exact match query or point query
- Range searching
- kNN searching
  - kNN searching order

General framework for hierarchical object-based index structures

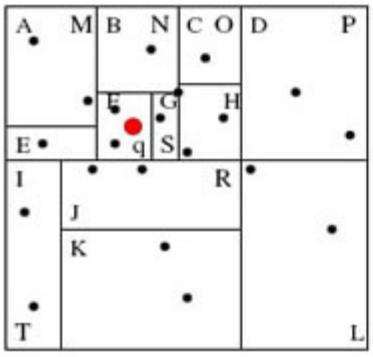
- Objects bounded by a hyper-dimensional bounding rectangle or bounding box or bounding sphere or bounding convex polygon
- Total of N objects N
- $\mathbf{M}$ objects of one level t into a bigger object of the next level
- Bigger object D_i at level i encompasses all smaller objects D_{i+1} at level $i+1$
- Total number of bounding objects is

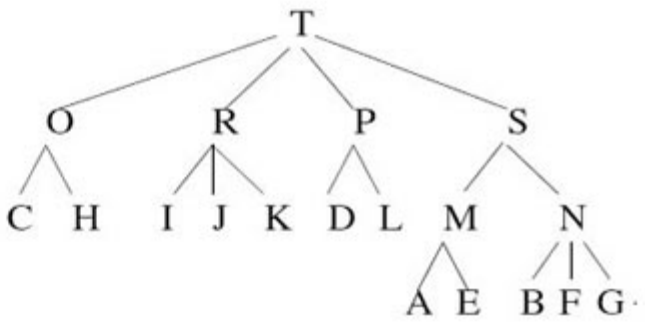
$$\lceil N/M \rceil + \lceil N/M^2 \rceil + \cdots = O(N/M)$$

- Total number of objects is, therefore, still $O(N)$
- Hierarchy is called object pyramid
- Provides multi-resolution representation of database objects

Exact match query or point query

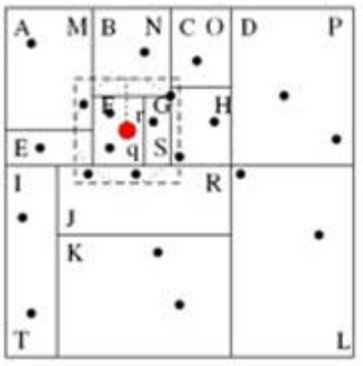
- Query point q
  - Start with level 0
  - At level i, determine if q is in object D_i
    - If no, return
    - If yes, recurse for all child objects in D_{i+1}
- Example
  - q is in T, so recurse
  - q not in O, R, P but in S
  - Proceed search to S
- May follow multiple paths when objects are non-disjoint
- May end up doing more work than sequential scan

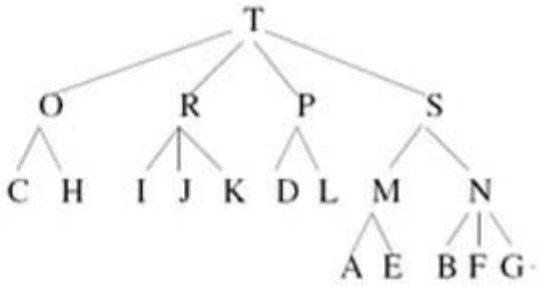




Range searching

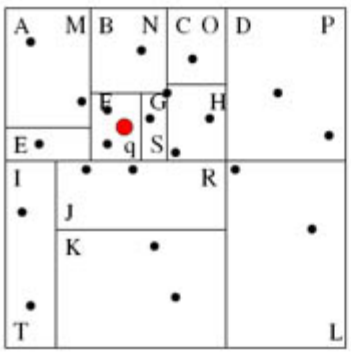
- Query point q and a distance radius r
- Can be found by region intersection
  - Region C is unnecessarily searched
  - Extra space increases with dimensionality
- Compute minimum and maximum distances to regions
  - If minimum $> r$ (C,D,I,L,K), no point in answer set-prune
  - If maximum $\leq r$ (F), all points in answer set-report
  - Otherwise (A,B,E,G,H,J), not sure-recurse into the region

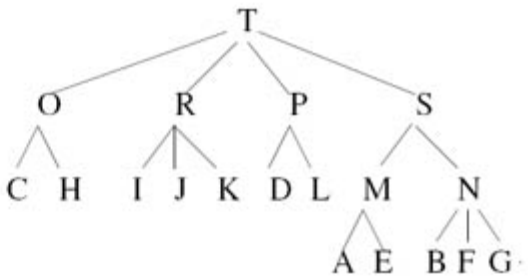




kNN searching

- Query point q and a positive integer k
- Maintain heap of k current estimates and k^{th} distance d_k
- Compute minimum and maximum distances to regions
  - If minimum $> d_k$, no point in answer set { prune
  - If maximum $\leq d_k$ and region contains at most k points, all points in answer set- insert into heap
  - Otherwise, not sure-recurse into the region





kNN searching order

- Efficiency depends on order in which regions are searched
  - Suppose k = 2
  - Starting with F results in better pruning than with L
  - For root T, $\langle$S,O,R,Pi$\rangle$ order is better than $\langle$P,R,O,Si$\rangle$ order
- Start with the region with lowest minimum
  - Sorting or comparison among regions
- Maintenance of heap
- More complex and time-consuming than range querying
  - Eorts to estimate range r for k neighbours