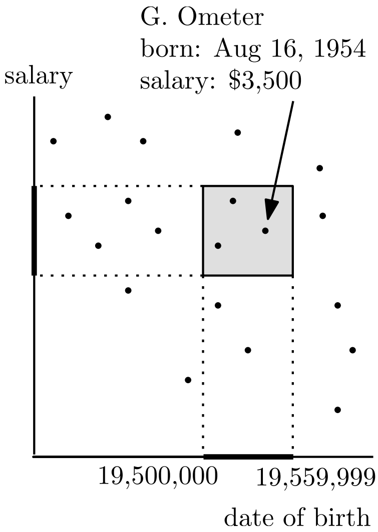
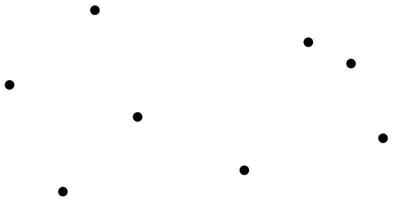
Database queries

A database query may ask for all employees with age between a_1 and a_2, and salary between s_1 and s_2



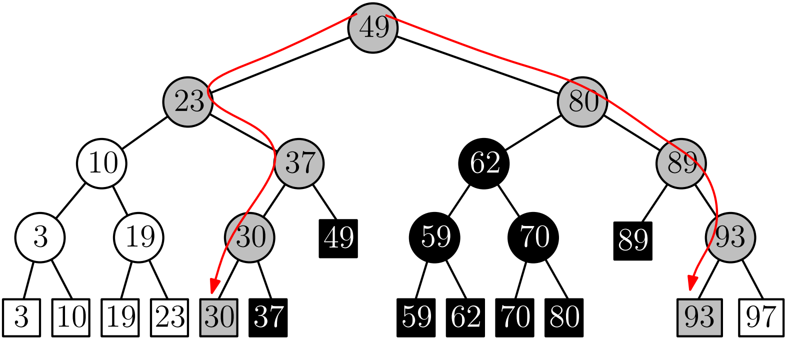
Faster queries

Can we achieve $O(\log n\ [+k])$ query time?



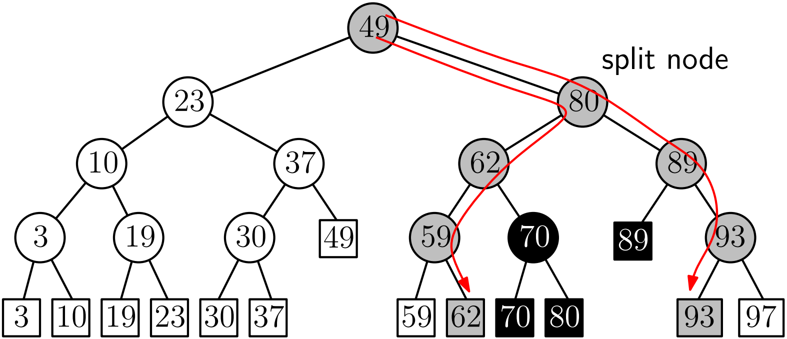
Example 1D range query

A 1-dimensional range query with $[25, 90]$



Example 1D range query

A 1-dimensional range query with $[61, 90]$



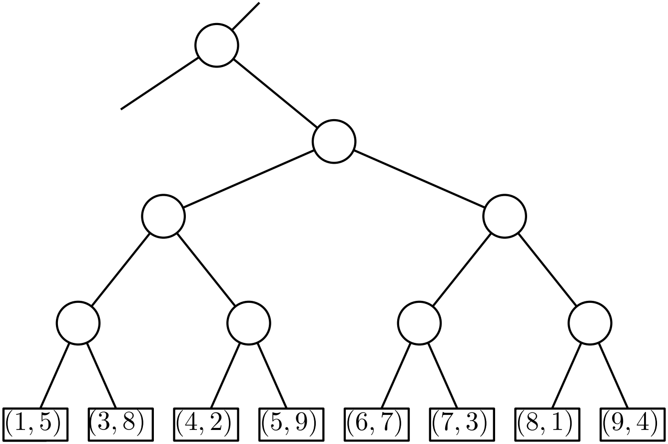
Examining 1D range queries

For any 1D range query, we can identify $O(\log n)$ nodes that together represent all answers to a 1D range query

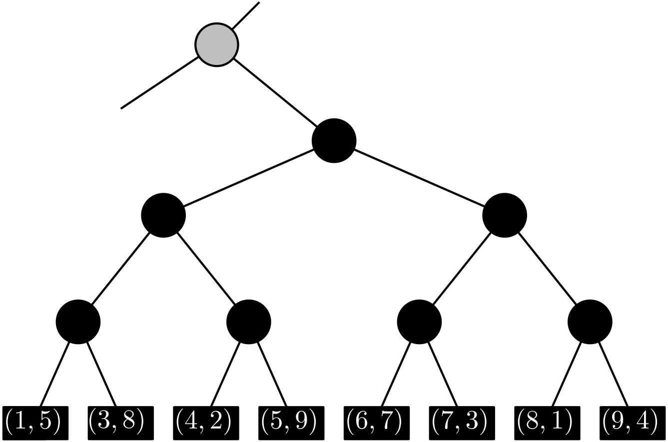
Toward 2D range queries

For any 2d range query, we can identify $O(\log n)$ nodes that together represent all points that have a correct first coordinate

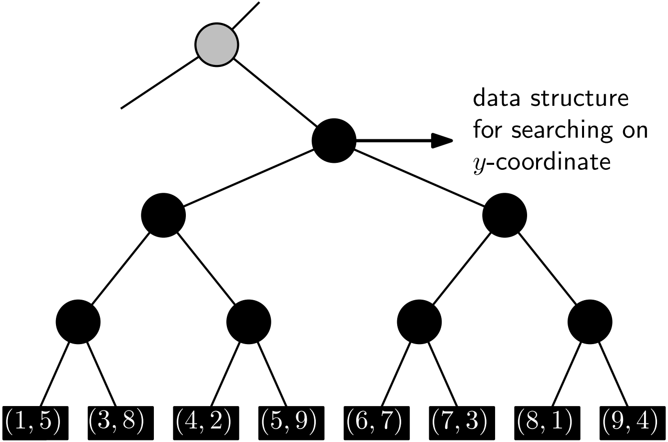
Toward 2D range queries



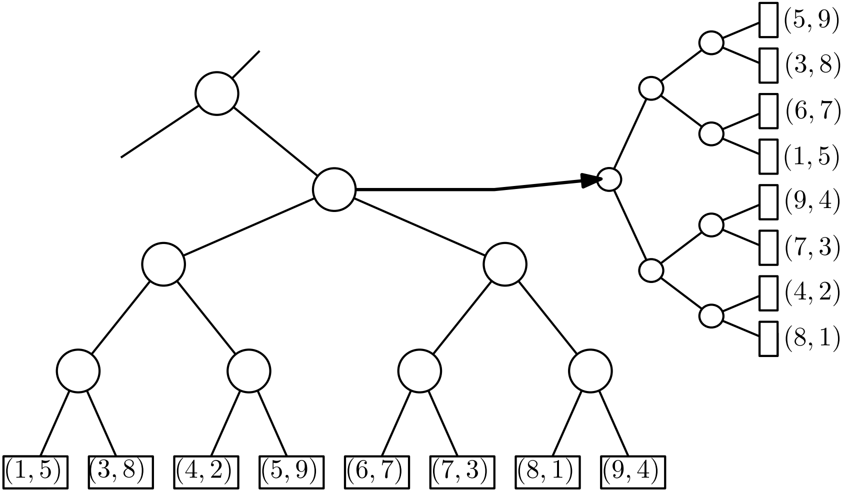
Toward 2D range queries



Toward 2D range queries

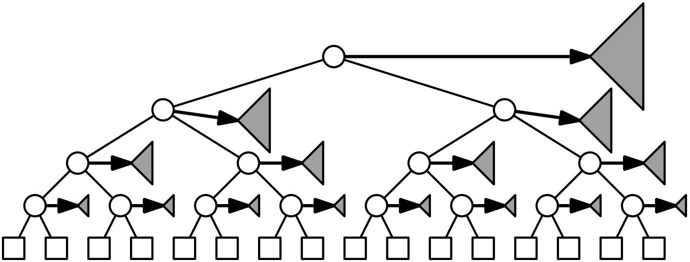


Toward 2D range queries



2D range trees

Every internal node stores a whole tree in an *associated structure*, on y-coordinate



Question: How much storage does this take?

Result

Theorem: A set of n points in the plane can be preprocessed in $O(n \log n)$ time into a data structure of $O(n \log n)$ size so that any 2D range query can be answered in $O(\log^2 n + k)$ time, where k is the number of answers reported

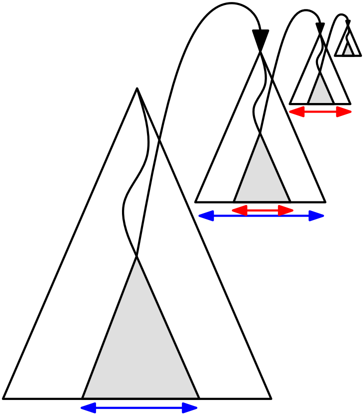
Recall that a kd-tree has $O(n)$ size and answers queries in $O(\sqrt{n} + k)$ time

Efficiency

n	$\log n$	$\log^2 n$	$\sqrt{n}$
16	4	16	4
64	6	36	8
256	8	64	16
1024	10	100	32
4096	12	144	64
16384	14	196	128
65536	16	256	256
1M	20	400	1K
16M	24	576	4K

Higher dimensional range trees

A d-dimensional range tree has a main tree which is a one-dimensional balanced binary search tree on the first coordinate, where every node has a pointer to an associated structure that is a $(d-1)$-dimensional range tree on the other coordinates



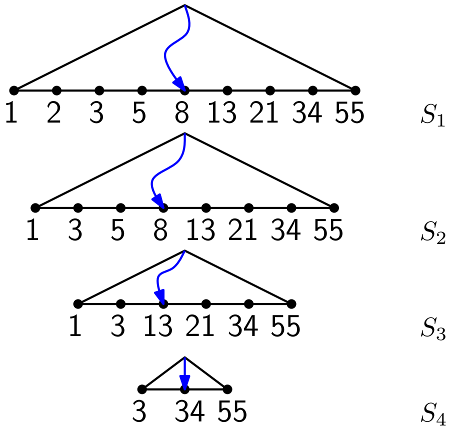
Improving the query time

The idea illustrated best by a *different* query problem:

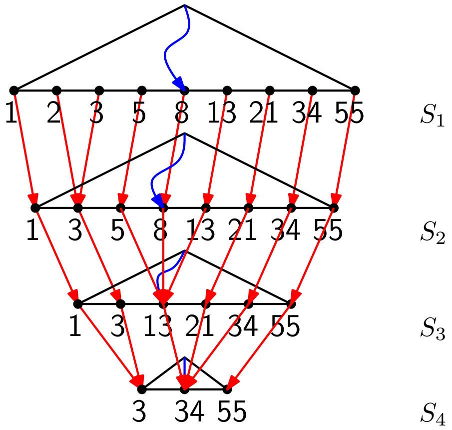
Suppose that we have a collection of sets $S_1, \ldots, S_m$, where $|S_1| = n$ and where $S_{i+1} \subseteq S_i$

We want a data structure that can report for a query number x, the smallest value $\geq x$ in all sets $S_1, \ldots, S_m$

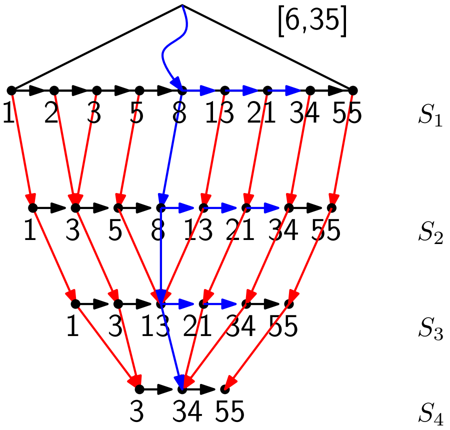
Improving the query time



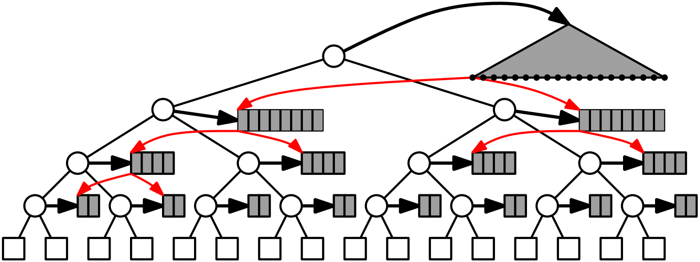
Improving the query time



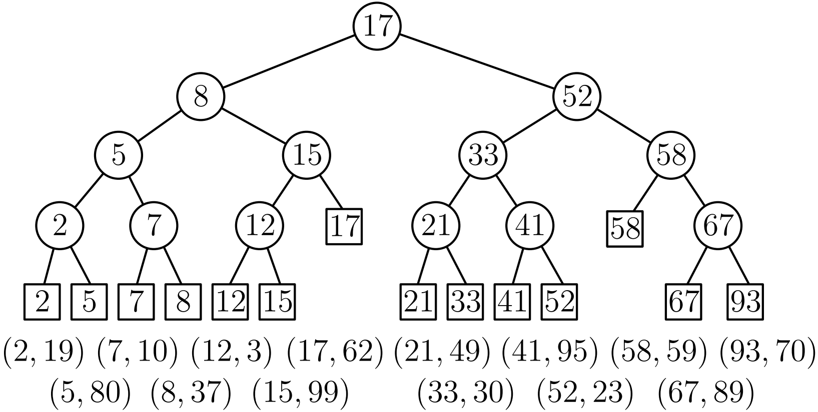
Improving the query time



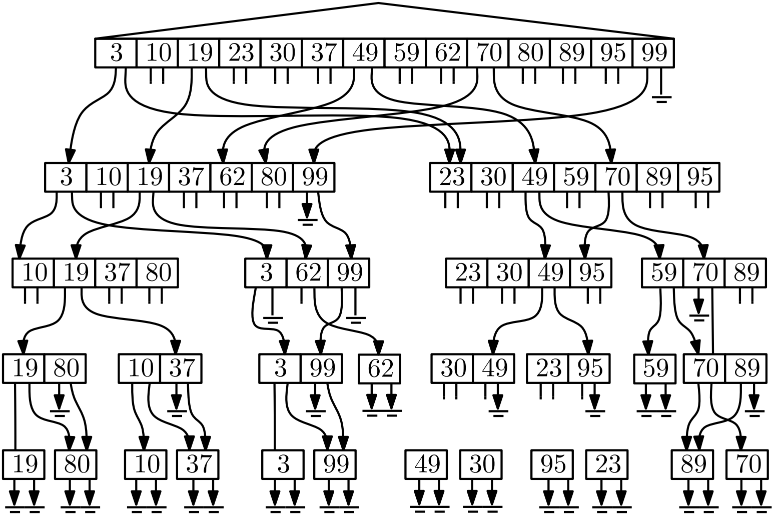
Fractional cascading



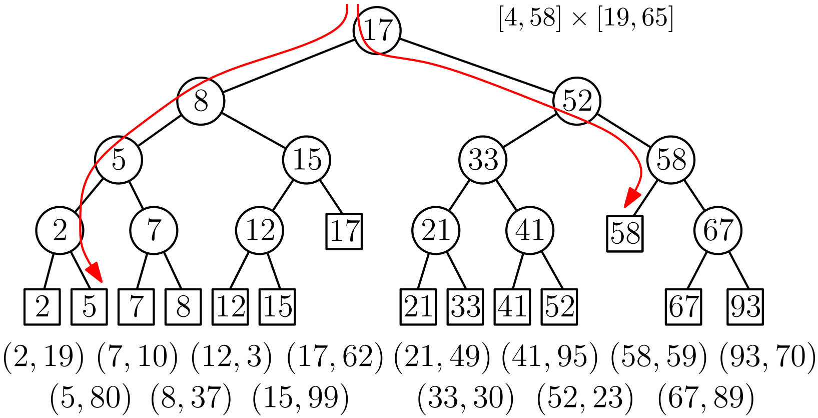
Fractional cascading



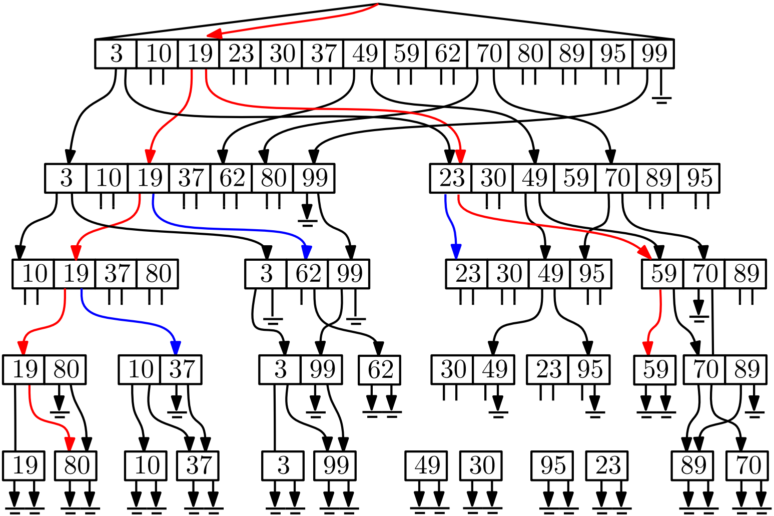
Fractional cascading



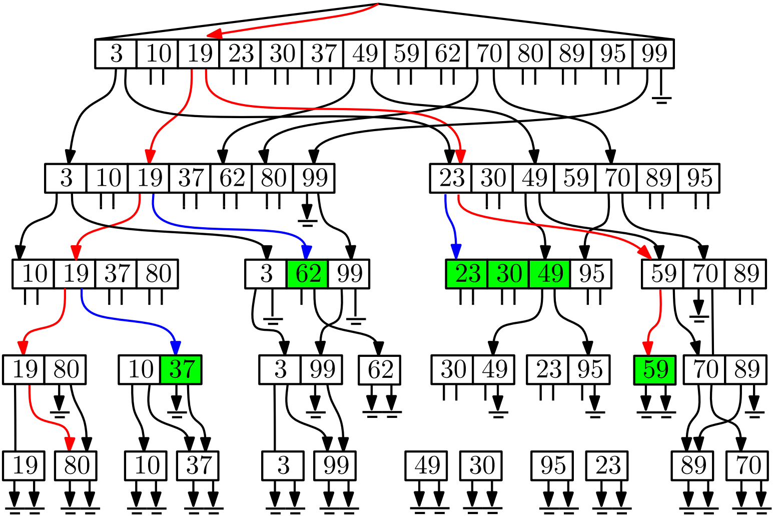
Fractional cascading



Fractional cascading



Fractional cascading



Result

Theorem: A set of n points in d-dimensional space can be preprocessed in $O(n \log^{d-1} n)$ time into a data structure of $O(n \log^{d-1} n)$ size so that any d-dimensional range query can be answered in $O(\log^{d-1} n + k)$ time, where k is the number of answers reported

Recall that a kd-tree has $O(n)$ size and answers queries in $O(n^{1-1/d} + k)$ time