

COP 5405: Advanced Algorithms

Fall 2006

Lecture 20

Layered range trees

We discuss a 2-dimensional layered range tree. Note that $P(\text{leftchild}(v)) \subseteq P(v)$ and $P(\text{rightchild}(v)) \subseteq P(v)$. So the fractional cascading idea can be used.

- The main tree is a balanced BST on the x coordinates.
- The associated data structure for node v in the above tree is a sorted array.
 - The array is sorted on the y coordinates.
 - Each array element stores two pointers, one to an array element in the associated data structure of $\text{leftchild}(v)$, and one to an array element in the associated data structure of $\text{rightchild}(v)$, based on the fractional cascading idea.
- A d -dimensional layered range tree can be constructed in time $O(n \log^{d-1} n)$ time and uses $O(n \log^{d-1} n)$ storage, $d \geq 2$.

A 2-dimensional range tree is queried as in 2-dimensional range tree, except that the 1-dimensional search is performed as follows.

- For array associated with v_{split} , perform binary search to find the location i corresponding to y .
- At a lower level node, find the location pointed to by $A[v_{\text{split}}](i)$.
- Repeatedly follow subsequent pointers at lower levels too.
- Query time is $O(\log^{d-1} n + k)$ to report k points, $d \geq 2$.