

In this lecture, the first of two about geometric data structures, we'll talk about two major problems—point location and range searching—and tie them to several major data structural techniques: persistence, retroactivity, dynamization of augmentation through weight balance, and fractional cascading.

In (planar) point location, we need to preprocess a planar map to support queries of the form "which face contains this point?" This problem is useful for figuring out where you clicked in a GUI, or for figuring out what city/country/etc. contain your GPS coordinates. The static (preprocessing) version of this problem can be solved optimally (in logarithmic time per query) by applying persistence to a classic sweep-line technique. The dynamic version, where the map can be modified, can be solved for orthogonal maps in optimal logarithmic time per operation using retroactive successor from Lecture 2.

In (orthogonal) range searching, we need to preprocess n points in d dimensions to support queries of the form "which points are in this box?" Simple range trees solve this problem in $O(\log^d n)$ query time and $O(n \log^{d-1} n)$ space. A cool crosslinking technique (called layered range trees) reduce the query time to $O(\log^{d-1} n)$. A general technique for dynamizing augmented search trees, called weight-balanced trees, allows us to additionally support insertion and deletion of points in $O(\log^d n)$ time per update. The crosslinking technique is a part of a more general technique called fractional cascading, which can further improve query time by another logarithmic factor.

MIT OpenCourseWare
<http://ocw.mit.edu>

6.851 Advanced Data Structures
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.