This lecture continues our theme of dynamic graphs. Beyond surveying results for several such problems, we'll focus on dynamic connectivity, where you can insert and/or delete edges, and the query is to determine whether two vertices are in the same connected component (i.e., have a path between them). We'll cover a few different results for this problem:

1. Link-cut trees (from last lecture) already solve trees in $O(\lg n)$.
2. Euler-tour trees are a simpler way to solve trees in $O(\lg n)$, which allow us to aggregate over subtrees instead of paths.
3. If we just insert or just delete edges, we can solve trees in $O(1)$, using our friend leaf trimming plus some simple bit-vector tricks.
4. For general graphs, we'll see how to support updates in $O(\lg^2 n)$ and queries in $O(\lg n / \lg \lg n)$.

This will be our culmination of data structures for dynamic graphs; the next (final) lecture is about matching lower bounds.

6.851 Advanced Data Structures
Spring 2012