

CS231 Fall 2013 Lecture 35

Stephanie R Taylor

November 25, 2013

1 Administrative Topics

- Yes, there is lab today. It is designed to help you hunt the rumpus.

2 Skip Lists

Skip lists are data structures that use randomness to maintain an ordered list of data with $O(\log n)$ insert, add, and removal operations. Skip lists aim to have the performance of a balanced binary search tree without the pain of balancing the tree. They are Stephanie's favorite data structure (no comments on grammar, please :)).

Because they use randomness, skip lists are **probabilistic** data structures. The order of insert, add, and removal operations is not guaranteed to be $\log n$, but there is a high probability that they will be.

The basic idea behind a skip list is to have a linked list, but, instead of giving each node just one pointer to its neighbor, it gives the node an array of pointers. The length of the array of pointers determines the “level” of the node. All nodes have at least one level. The pointers at the bottom level form a regular linked list. The pointers one level up chain together the nodes with at least 2 levels. The pointers one more level up chain together the nodes with at least 3 levels. To find a key in a skip list you follow the highest level pointers you can (taking huge strides through the list), until you would

overshoot your goal, then use pointers one level down, etc. until you find your key.

If the distribution of levels in the nodes is geometric (i.e. there are twice as many 1-level nodes as 2-level nodes) and the higher-level nodes are evenly spaced, then the data structure is most efficient. Guaranteeing these distributions is problematic (it would be expensive), so we tend to use random approaches and assume it will all work out. It usually does.