CSE 347 Analysis of Algorithms March 18, 2019

Lecture 15 Summary

Instructor: Brendan Juba

1. By making random choices, we can obtain algorithms that obtain a worst-case guarantee "with high probability," i.e., with probability greater than $1 - \delta$ for some $\delta > 0$.

2. If we insert elements into a binary search tree in random order, the worst-case depth is $O(\log \frac{n}{\delta})$ with probability $1 - \delta$.

   (a) Given the tree so far, we look at the elements that will fall into the same subtree as a fixed element x_i. The next element to be inserted (that will appear as an ancestor for this subtree) is in a uniform-random location among this range.

   (b) So, with probability $1/2$, the next element greater than x_i is closer to x_i than $1/2$ of the elements greater than x_i, in which case, the number of elements remaning reduces by $1/2$. This can only happen $\log_2 n$ times before no elements greater than x_i remain.

   (c) Since this occurs with probability $1/2$ independently each time we pick an element from this range, by a *Chernoff bound*, in $O(\log \frac{n}{\delta})$ trials, the probability that we have not yet elimintated all of these elements is at most $\frac{\delta}{2n}$.

   (d) Now, by a *union bound* over the corresponding events for $x_1, \ldots, x_n$, and the elements greater than and less than each x_i, we find that the total probability that any x_i has depth greater than $O(\log \frac{n}{\delta})$ is at most δ, as needed.

3. For some large N, if we are making fewer than $O(\sqrt{N \log \frac{1}{1-\delta}})$ inserts, then we can obtain $O(\log \frac{n}{\delta})$ time inserts with probability $1 - \delta$ as well

   (a) We assign each element a uniform-random "priority" from the range O–N, and maintain the invariant that the elements in the tree have increasing priorities from root-to-leaf using rotations

      i. We insert normally, and rotate the new element up the tree until its priority is greater than its immediate ancestor

      ii. Since this is the only element that may violate the priority relationships, the invariant will be maintained at that point

   (b) As long as there are no collisions among the priorities (true with probability $1 - \delta$ for $O(\sqrt{N \log \frac{1}{1-\delta}})$ by the calculation from last lecture) the priorities uniquely determine the tree, which is therefore the same tree we would obtain by inserting in order of increasing priority.

   (c) By the previous analysis, the tree therefore maintains depth $O(\log \frac{n}{\delta})$ with probability $1 - \delta$.

4. The sorting algorithm Quicksort has running time $O(n \log \frac{n}{\delta})$ with probability $1 - \delta$ since the recursion tree is a binary search tree on a random ordering of the elements.

(a) We label the calls with the selection of pivots, and label the two recursive calls the "left" and "right" children of this call to obtain the correspondence.

(b) Since the array is partitioned among the calls at each level, the total work at each call depth is $O(n)$, so multiplying by the total depth gives a total running time of $O(n \log \frac{n}{\delta})$ as claimed.