Assignment 3

Due: February 4, 2020

Kanyid, Bradon
`bradon.kanyid@pdx.edu`

Reimer, Daniel
`daniel.reimer@pdx.edu`

Problem 1: Lower Bounds [10 points] Prove that any comparison-based algorithm for constructing a binary search tree from an arbitrary list of n elements takes $\Omega(n \log n)$ time in the worst case. (Hint: Consider how to reduce the sorting problem to performing a set of operations on a binary search tree. In other words, show that if a faster algorithm existed for constructing a binary search tree then you would violate the $\Omega(n \log n)$ comparison-based sorting lower bound.)

Solution

We don't know how long any possible comparison-based BST construction algorithm would take. However, we do know that if we constructed a BST, we could reduce sorting to an in-order traversal of a BST. We also know that in-order traveral has $O(n)$-time complexity.

Assume that we could efficiently construct a BST in $o(n \lg n)$-time, i.e. faster than $O(n \lg n)$. Since sorting an unsorted list reduces to a BST construction based on that unsorted list, followed by in-order traversal, and we know that in-order traversal is $O(n)$-time complexity, then the complexity of sorting an unordered list would be $o(n \lg n)$.

However, this is a contradiction because we know sorting has $\Omega(n \lg n)$-time complexity. Therefore BST construction must also have $\Omega(n \lg n)$-time complexity.

Problem 2: Median of Medians The 'Median-of-medians' selection algorithm presented in class divides the input into groups of 5. Using a group of odd size helps keep things a little simpler (because otherwise the group medians are messier to define), but why the choice of 5?

(a) [10 points] Show that the same argument for linear worst-case time complexity works if we use groups of size 7 instead.

Solution

We divide the elements with a recursive call on $n/7$. We are able to partition the groups with 4/7 items smaller than the pivot (this is $(1/2)*4*(n/7) == 4n/14$ groups), and groups with 4/7 items larger than the pivot (this is also $(1/2)*4*(n/7) == 4n/14$). These are the known elements we can eliminate, which leaves us with $5n/7$ elements to recurse on and further filter.

$$\begin{aligned}
T(n) &\leq an + T(\frac{n}{7}) + T(\frac{5n}{7}) \\
&= an * \sum_{i=0}^{\infty} \frac{6n}{7} \\
&= an * \frac{1}{(1 - (\frac{6}{7}))} \\
&= 7an \in O(n)
\end{aligned}$$

With splitting into groups of 7, we get a lower constant but the overall performance is still $O(n)$

(b) [10 points] Show that groups of size 3 results in superlinear time complexity.

Solution

We divide the elements with a recursive call on $n/3$. We are able to partition the groups with 2/3 items smaller than the pivot (this is $(1/2)*2*(n/3) == 2n/6$ groups), and groups with 2/3 items larger than the pivot (this is also $(1/2)*2*(n/3) == 2n/6$). These are the known elements we can eliminate, which leaves us with $2n/3$ elements to recurse on and further filter.

$$\begin{aligned}
T(n) &\leq an + T(\frac{n}{3}) + T(\frac{2n}{3}) \\
&= an * \sum_{i=0}^{\infty} \frac{3n}{3} \\
&= an * n \\
&= an^2 \in O(n^2)
\end{aligned}$$

Because we are recursing on the tree with $n/3$ groups and $2/3n$ groups, the combined recursion ends up looking at n total elements, i.e. the recursion is not reducing the problem.