Recitation 7

Combining BSTs

7.1 Announcements

- *FingerLab* is due **Friday afternoon**. It's worth 125 points.
- *RangeLab* will be released on **Friday**.

7.2 Generalized Combination

In lecture, we discussed `union`, and argued that it has $O\left(m \log\left(\frac{n}{m} + 1\right)\right)$ work and $O(\log(n)\log(m))$ span. The latter bound can be improved to $O(\log n + \log m)$ using *futures*[1], but that is outside the scope of this course.

What about the functions `intersection` and `difference`? These can be implemented in a similar fashion as `union`, and as such have the same cost bounds. In this recitation, we'll establish this more concretely.

Task 7.1. *Implement all three functions* `union`, `intersection`, *and* `difference` *in terms of a single helper function* `combine` *which has* $O\left(m \log\left(\frac{n}{m} + 1\right)\right)$ *work and* $O(\log(n)\log(m))$ *span for BSTs of size* n *and* m, $n \geq m$. *Conclude that all three of these functions have the same cost bounds.*

Task 7.2. *Consider a function* `symdiff` *where* (`symdiff` (A, B)) *returns a BST containing all keys which are either in* A *or* B, *but not both. Implement* `symdiff` *in terms of* `combine`.

[1] http://dl.acm.org/citation.cfm?id=258517