6.897
L3.2
Feb. 8, 2005

ADT/DS interface:

- [Insert/Delete] } essentially the same in this model
- [Successor/Predecessor] }
- Search } focus on this & on successful searches

Model of computation (& costs): (there are several, equivalent)

- n keys in n nodes (not just leaves)
- single pointer to node of BST ("finger")
- resets to root at start of each ~~search~~ access
- operations: left, right, parent
  rotate (this node & its parent)
- must at some time point to answer of query

$\lg n$ worst case lower bound:

access deepest node which must be depth $\geq \lg n$.

Static OPT

- suppose OPT can't rotate
- dynamic program to compute this OPT: ← OFFLINE ALGORITHM – know x in advance
  - guess which element r is the root — n choices
  - get two subproblems
  - cost/access $= 1 + \Pr[<r] \cdot$ cost(left subproblem) $+ \Pr[>r] \cdot$ cost(right subproblem)
  - all about f_r = frequency(r) = # occurrences of r in x
  - $O(n^2)$ subproblems corresp. to contiguous interval of elements

  $\Rightarrow O(n^3)$ time
- smarter DP: $O(n^2)$ time [Knuth 1971]

  OPEN: faster?
  - possible ~~that~~ if solution can be approximate
- static OPT/access is between $H - \lg H - \lg e + 1$ and $H + 3$

  where $H = \text{entropy} = \sum_i p_i \lg 1/p_i$, and $p_i = f_i/m$
- BST is statically optimal (or has entropy bound) if its cost = O(static OPT cost) = O(entropy)
  - cool if you can achieve without knowing x a priori — ONLINE ALGORITHM