MA/CSSE 473 Day 30 Announcements and Summary

Announcements:

1. **HW 12** due Thursday, Oct 30.
2. **I will be off-campus** Oct 30 in the afternoon and most of Oct 31 (I hope to be here for hours 9-10) due to my IVIG infusions.
3. **No class meeting Oct 31.**
4. **Exam 2** Tuesday Nov 4 in class. Exam specification is linked from Day 34 in the schedule page.
5. **HW** 13 due Thursday, Nov 6, HW 14 Monday Nov 10.
6. **Final Exam Monday Nov 17 at 6:00 PM.**
7. **In my office today**: hours 6, 8, 9.

Main ideas from today: Optimal Binary Search trees.

1. Formally, an Extended Binary Tree (EBT) is either
   a. an external node, or
   b. an (internal) root node and two EBTs T_L and T_R
2. Optimal BST notation:
   a. Keys are $K_1, K_2, \ldots, K_n$
   b. Let v be the value we are searching for
   c. For i= 1, …,n, let a_i be the probability that v is key K_i
   d. For i= 1, …,n-1, let b_i be the probability that $K_i < v < K_{i+1}$
   e. Similarly, let b_0 be the probability that $v < K_1$, and b_n the probability that $v > K_n$

      $$\sum_{i=1}^{n} a_i + \sum_{i=0}^{n} b_i = 1$$

   f. We can also just use *frequencies* instead of *probabilities* when finding the optimal tree (and divide by their sum to get the probabilities if we ever need them). That is what we will do.

3. We want to minimize weighted path length,

   $$C(T) = \sum_{i=1}^{n} a_i[1 + depth(x_i)] + \sum_{i=0}^{n} b_i[depth(y_i)]$$

4. You will show by induction (HW 12) that C(T) can be calculated by the recursive formula
   a. C(empty EBT) = 0,
   b. If T has a root and two subtrees T_L and T_R, $C(T) = C(T_L) + C(T_R) + \Sigma a_i + \Sigma b_i$,
   c. where the summations are over all a_i and b_i for nodes in T

5. Consider these Frequencies of vowel occurrence in English A, E, I, O, U
   a's: 32, 42, 26, 32, 12
   b's: 0, 34, 38, 58, 95, 21

6. Define the quantities W_{ij} that help with the calculation of the C_{ij}.

7. R_{ij} (an integer) is the index of the best key to use as a root of the optimal tree.

   It is the value of k that minimizes

8. What is the running time of the optimalBST algorithm, as a function of the number of keys?

9. What do we mean by "greedy" algorithm?