Cosi 30b Quiz #2 Solutions 3 April, 1997

1. (a) (10pts) $T(1) = 0;\ T(n) = T(n-1) + n :\ T(n) = O(n^2)$
   We prove $\exists c > 0 :\ \forall n \geq 1\ T(n) \leq cn^2$.
   Defer the basis since we haven't chosen c yet. Assume the claim true for all $m < n$; then $T(n) \leq c(n-1)^2 + n = cn^2 - 2cn + c + n \leq cn^2$ for all $n \geq 1$ and $c = 1$. Now we know c and can prove the basis.

   (b) (10pts) $T(1) = 0;\ T(n) = 2T(n/2) + n :\ T(n) = O(n \lg n)$ for n a power of 2.
   Assume $T(n/2) \leq c(n/2)\lg(n/2)$. Then $T(n) \leq 2c(n/2)(\lg n - 1) + n = cn \lg n - cn + n$. So we can choose $c = 1$ and also prove the basis.

   (c) Note the following:

   i. Get rid of the "big-Oh" notation *before* starting the proof.

   ii. Avoid introducing extraneous coefficients beyond the single necessary c.

   iii. O, Θ, Ω cannot be added, multiplied, or transposed in equations. They are not *values*, they are shorthand for complicated logical statements — "There exist constants c and n_0 such that for all $n > n_0$..."

   iv. The constant c in the conclusion must be the same as the constant in the assumption. Otherwise you can prove falacies, such as $n = O(1)$: Basis $1 = 1 = O(1)$. Assume $n - 1 \leq c$. Then $n = n - 1 + 1 \leq c + 1 = c' = O(1)$.

2. Let a hash table have m slots.

   (a) (5pts) Assume the contrary, that each slot holds $d - 1$ or fewer keys. Then the total number of keys stored is at most $m(d-1) < md$.

   (b) (5pts) There are $n(n-1)/2$ ordered pairs. By the randomness assumption, each pair independently has probability $1/m$ of colliding, so the expected number of collisions is $n(n-1)/2m$.

   (c) (i)(5pts) There are m choices of which list to fill and exactly 1 way that every key hashes there. Probability $= m \times m^{-n}$

   (ii, iii) (not graded) The next two were too difficult for a quiz. Some students came close by counting the number of ways of choosing i empty lists in $\binom{m}{i}$ ways, then putting the n keys in the remaining $m - i$ slots in $(m-i)^n$ ways, then summing from i to m. But that double-counts some configurations: once when the list is empty because it was chosen to be, and again when it is empty "by accident." So for these two subproblems you need to express the probability that *no* list among a set is empty, and for this you need to use Venn diagrams or the "inclusion-exclusion" rule.

   (d) (5pts)

   ```
   [ /, /, 1, 5, 8, 13, 15 ]
   ```

3. Search Trees

   (a) (6pts) Two solutions were accepted for the deletion: The answer depends upon whether you assume the procedure is passed the root of the tree or a pointer to the root.

   (b) (3pts) Yes, unlike the homework problem, this one asked about the search key itself. By definition of a BST, everything to the left is $< k$.

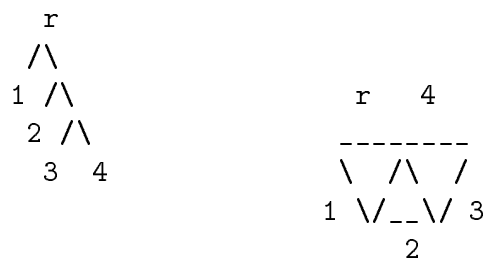
   (c) (4+2pts) Two answers acceptable: Either because building the tree plus a pre-order traversal achieves comparison sorting, or by a direct analysis yielding the sum $\sum i \lg i$. The better bound is $\Omega(n^2)$

   (d) (5pts) No.

   (e) (3pts) Space is the reason: the 2,3,4 structure may allocate unused pointers, while the red-black tree uses all it allocates.

4. Dynamic Programming

   (a) (10pts) Standard proofs use induction, dividing the polygon either by a single chord or by the triangle on edge (0,n+1) used in associating the triangulation with a full binary tree. Then count the chords and triangles to the left and to the right. Notice that you can't simply add a vertex and two new edges to an (n-1) polygon without arguing first that every full triangulation *has* such a triangle.

   (b) (4+1pts) In a "full" binary tree (p. 95) every node has either 0 or 2 children. . The easiest argument is that $T(n) = \sum_k T(k)T(n-k-1) \geq \sum_k T(k)$, which gives $T(n) = \Theta(2^n)$ but not $\Theta(n^3)$.

   (c) (10pts) The dynamic program had row values (from bottom up) of 24, 60, 15; 54, 27; 35. The association was (1(2(34))). The tree and polygon are shown below: (3+2pts)