CSCI 3100: Algorithms

Homework 1

Required Problems

1. Solve the following recurrences. State tight asymptotic bounds for each function in the form $\Theta(f(n))$ for some recognizable function $f(n)$. You do not need to turn in proofs (in fact, please *don't* turn in proofs), but it's a good idea to work through these for practice. Assume reasonable but nontrivial base cases if none are supplied. More exact solutions are better.

   (a) $A(n) = 4A(n/2) + n$

   (b) $B(n) = 2B(3n/4) + n^2$

   (c) $C(n) = C(n/3) + n \log n$

   (d) $D(n) = 3D(n-1) + 1$

   (e) $E(n) = E(n-1)^2 - E(n-1)^2$, where $E(0) = 0$ and $E(1) = 1$.
   [Hint: This is easier than it looks! Try writing out a few terms and see what happens.]

2. You are a visitor at a political convention with n delegates; each delegate is a member of exactly one political party. It is impossible to tell which political party any delegate belongs to; in particular, you will be summarily ejected from the convention if you ask. However, you can determine whether any pair of delegates belong to the same party by introducing them to each other. Members of the same political party always greet each other with smiles and friendly handshakes; members of different political parties always greet each other with angry stares and insults.

   Suppose more than half the delegates belong to the same political party. Describe an efficient algorithm that identifies all members of this majority party. If you write pseudocode (which I highly recommend) be sure to formally define its input parameters and any subroutines you use.

   [Hint: Write a divide-and-conquer algorithm. Any correct algorithm with a running time of O(n log n) is worth full credit - but (as always) you have to prove correctness and the running time.]

3. Recall that a binary tree is full if every non-leaf node has exactly two children. Describe and analyze a recursive algorithm to reconstruct an arbitrary full binary tree, given its preorder and inorder node sequences as input. (Assume all keys are distinct in the binary tree).

   [Hint: You did this on the last homework!]

4. Sample solved problem - to be passed out in class! Come see me if you miss class and need to pick up a copy.