CSE 101
Practice Midterm: Oct. 18
Topics: Order, Recurrence Relations, Analyzing Programs,
Divide-and-Conquer, Back-tracking
Time: 50 min.

Order: 25 points. Prove that for any constants $a, b > 0, k \geq 1$, $(bn + a)^k \in \Theta(n^k)$.

Divide-and-Conquer Recurrence: 25 points Consider the following recursive algorithm. It's inputs are two *sorted* n element arrays A and B (so the algorithm assumes $A[1] < A[2] < ... < A[n]$ and $B[1] < B[2] < ... < B[n]$). The algorithm returns the *median* of the two lists combined, i.e., the n'th smallest out of all $A[i]$ and $B[j]$'s. Give a recurrence relation for the time the algorithm takes and use it to give a time analysis up to order.

1. Program: SortedMedian(A[1..n], B[1..n]: Arrays of Integers): Integer;
2. IF $n = 1$ THEN return $Min(A[1], B[1])$.
3. $k \leftarrow \lceil n/2 \rceil$
4. $k' \leftarrow \lceil (n+1)/2 \rceil$
5. IF $A[k] \geq B[k]$ THEN return $SortedMedian(A[1...k], B[k'...n])$
6. ELSE return $SortedMedian(A[k'..n], B[1..k])$

Binary Tree Divide-and-Conquer: 25 points Consider the following problem. The input is a binary tree data structure (each node has parent pointer $Parent$, right-child pointer $Right$, and left-child pointer $Left$). If no right-child or left-child exists, the pointers return "NIL". In addition, each node is labelled with a numerical value, $Val(x) \geq 0$. Let T_x be the sub-tree rooted at x. We want to label each node with a new field $Max(x) = Max_{v \in T_x} Val(v)$.

Present (as a pseudo-code program or in precise English) and analyze a linear time ($O(n)$) divide-and- conquer algorithm to compute this new field for all nodes x in the tree.

Back-Tracking Consider the following problem. The input is a list of n classes $C_1, ..C_n$ you need to take to graduate in June; each must be taken in one of this year's three quarters, Fall, Winter, or Spring. You also have as input a list of conflicts between classes, each conflict specifying two classes and two quarters, e.g. a conflict between CSE 101 in the Fall and CSE 100 in the Winter (because CSE 100 is a prereq for CSE 101) or between Math 103 in the Fall and CSE 141 in the Fall (because they are being taught at the same time.) A. (20 pts) Describe a back-tracking algorithm

to determine whether it is possible to schedule all n courses this year without creating a conflict. You can use either pseudo-code or a clear English description. B. (5 points) Use your algorithm to schedule the following example: Courses: CSE 100, CSE 101, Parapsychology 200, and World Domination 101. There are conflicts between:

1. CSE 100 in the Fall and CSE 101 in the Fall (prerequisite)
2. CSE 100 in the Winter and CSE 101 in the Fall and Winter
3. CSE 100 in the Spring and CSE 101 in the Fall, Winter and Spring
4. CSE 100 in the Winter and both World Domination 101 and Parapsychology 200 in the Winter (overlapping times)
5. CSE 101 and World Domination 101 in all three quarters (too much work to do both!)
6. CSE 101 in the Spring and Parasychology 200 in the Spring (overlapping times)
7. Parapsychology 200 and World Domination 101 in all three quarters (they're always given at the same time)