CSE 551 Foundations of Algorithms
Sample Quiz
Closed Books, Closed Notes
Time: 1 hour
Each question carries 20 pts.

Problem 1: Algorithm A_1 takes $10^{-4} \times 2^n$ seconds to solve a problem instance of size n and Algorithm A_2 takes $10^{-2} \times n^3$ seconds to do the same on a particular machine.
(i) What is the size of the largest problem instance A_1 will be able solve in one year ?
(ii) What is the size of the largest problem instance A_1 will be able solve in one year on a machine one hundred times as fast ?
(iii) What is the size of the largest problem instance A_2 will be able solve in one year ?
(iv) What is the size of the largest problem instance A_2 will be able solve in one year on a machine one hundred times as fast ?
(v) Which algorithm will produce results faster, in case we are trying to solve problem instances of size less than 20 ?

Problem 2: Solve the following recurrence relation to find its asymptotic upper and lower bounds.
(i) $T(n) = 2T(n/2) + n$ for $n > 2$, and ($T(2) = 1$, for simplicity you may assume $n = 2^k$
Show all your work.

Problem 3: Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove of disprove the following conjectures:
(i) 3^n is not $O(2^n)$.
(ii) $log\ 3^n$ is $O(log\ 2^n)$.
(iii) n^n is not $O(n!)$
(iv) For $\alpha > 1$, $n^{\alpha log\ n}$ is not $O(n^{log\ n})$
(v) For $\alpha > 1$, $log\ n^{\alpha log\ n}$ is $O(log\ n^{log\ n})$

Problem 4: Given a set of numbers (a_1,...,a_n) develop an algorithm to find the largest and the second largest number in the set with at most $n + log\ n - 2$ comparisons. Analyze your algorithm to prove that the algorithm never requires more than $n + log\ n - 2$ comparisons. For simplicity, you may assume that n is a power of 2.