Five questions, each question 20 marks for a total of 100 marks. Due: on March 12, before 2:00 PM. Note: no extensions can be given because we will go through the solutions in class on that date.

1. **[20 marks]** You're given an array A of n integers, and must answer a series of n queries, each of the form: "How many elements a of the array A satisfy $L_k \leq a \leq R_k$?", where L_k and R_k $(1 \leq k \leq n)$ are some integers such that $L_k \leq R_k$. Design an $O(n \log n)$ algorithm that answers all of these queries.

2. **[20 marks, both (a) and (b) 10 marks each]** You are given an array S of n integers and another integer x.

   (a) Describe an $O(n \log n)$ algorithm (in the sense of the worst case performance) that determines whether or not there exist two elements in S whose sum is exactly x.

   (b) Describe an algorithm that accomplishes the same task, but runs in $O(n)$ **expected** (i.e., average) time.

   Note that brute force does not work here, because it runs in $O(n^2)$ time.

3. **[20 marks, both (a) and (b) 10 marks each; if you solve (b) you do not have to solve (a)]** You are at a party attended by n people (not including yourself), and you suspect that there might be a celebrity present. A *celebrity* is someone known by everyone, but does not know anyone except themselves. You may assume everyone knows themselves.

   Your task is to work out if there is a celebrity present, and if so, which of the n people present is a celebrity. To do so, you can ask a person X if they know another person Y (where you choose X and Y when asking the question).

   (a) Show that your task can always be accomplished by asking no more than $3n - 3$ such questions, even in the worst case.

   (b) Show that your task can always be accomplished by asking no more than $3n - \lfloor \log_2 n \rfloor - 2$ such questions, even in the worst case.

4. **[20 marks, each pair 4 marks]** Read the review material from the class website on asymptotic notation and basic properties of logarithms, pages 38-44 and then determine if $f(n) = \Omega(g(n))$, $f(n) = O(g(n))$ or $f(n) = \Theta(g(n))$ for the following pairs. Justify your answers.

$f(n)$	$g(n)$
$(\log_2 n)^2$	$\log_2(n^{\log_2 n}) + 2\log_2 n$
n^{100}	$2^{n/100}$
$\sqrt{n}$	$2^{\sqrt{\log_2 n}}$
$n^{1.001}$	$n \log_2 n$
$n^{(1+\sin(\pi n/2))/2}$	$\sqrt{n}$

   You might find the following inequality useful: if $f(n), g(n), c > 0$ then $f(n) < c\, g(n)$ if and only if $\log f(n) < \log c + \log g(n)$.

5. **[20 marks, each recurrence 5 marks)** Determine the asymptotic growth rate of the solutions to the following recurrences. If possible, you can use the Master Theorem, if not, find another way of solving it.

   (a) $T(n) = 2T(n/2) + n(2 + \sin n)$

   (b) $T(n) = 2T(n/2) + \sqrt{n} + \log n$

   (c) $T(n) = 8T(n/2) + n^{\log n}$

   (d) $T(n) = T(n-1) + n$