These problems are sample problems for the midterm exam, so you may expect similar problems in the midterm. Do not hand in your solutions. Solutions will be handed out and discussed (and posted on the web) on Monday, the day before the midterm. The midterm exam is a closed book exam, but you may use one sheet of paper (written on both sides) with notes on it. Each problem is worth 20 points.

1. Use the Master Theorem to find the asymptotic solutions for the following recurrences: (a) $T(n) = 7T\left(\frac{n}{2}\right) + n^2$, (b) $T(n) = T\left(\frac{n}{2}\right) + 1$, (c) $T(n) = 4T\left(\frac{n}{2}\right) + n^3$.

2. Use indicator random variables to find the expected value of the number of fixed elements (elements left in the same position) in a random permutation of $n$ elements.

3. We have two input arrays, an array $A$ with $m$ elements and an array $B$ with $n$ elements, where $m \leq n$. There may be duplicate elements. We want to decide if every element of $B$ is an element of $A$. Describe an algorithm to solve this problem in $O(n \log m)$ worst-case time.

4. We have an input array $A$ with $n$ ($n \geq 2$) numbers.
   (a) Describe a $O(n)$ worst-case time algorithm to find two elements $x, y \in A$ such that $|x - y| \geq |u - v|$ for all $u, v \in A$.
   (b) Describe a $O(n \log n)$ worst-case time algorithm to find two elements $x, y \in A$ such that $|x - y| \leq |u - v|$ for all $u, v \in A$.

5. The Fibonacci numbers are defined by the recurrence:
   
   $F_0 = 0, F_1 = 1, F_i = F_{i-1} + F_{i-2}$ for $i \geq 2$.

   Give an $O(n)$-time dynamic programming algorithm to compute the $n$th Fibonacci number. Draw the subproblem graph. How many vertices and edges are in the graph?