ESO207A: Data Structures and Algorithms	
Homework *1: Divide and Conquer*	HW Due Date: Aug 21, 2018

Instructions.

1. Start each problem on a new sheet. For each problem, write your name, Roll No., the problem number, the date and the names of any students with whom you collaborated. Remember that you must write the answer and the algorithm in your own words.
2. For questions in which algorithms are asked for, first summarize the problem you are solving and your results (including time/space complexity as appropriate). The body of the write-up should provide the following:
   (a) A clear description of the algorithm in English and/or pseudo-code, where, helpful.
   (b) At least one worked example or diagram to show more precisely how your algorithm works.
   (c) A proof/argument of the correctness of the algorithm.
   (d) An analysis of the running time of the algorithm.

   Remember, your goal is to communicate. *Full marks will be given only to correct solutions which are described clearly.* Convoluted and unclear descriptions will receive *low marks.*

Review and some definitions. Assume f, g to be asymptotically non-negative functions. In brief, $f(n) = O(g(n))$ if asymptotically, $f(n) \leq c \cdot g(n)$, for some constant c. $f(n) = \Theta(g(n))$ if asymptotically, $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$, for some constants c_1, c_2; that is, up to constant factors, $f(n)$ and $g(n)$ are asymptotically similar. $f(n) = \Omega(g(n))$ if asymptotically, $f(n) \geq c \cdot g(n)$, for some constant c. (Informally, asymptotically and up to constant factors, f is at least g). $f(n) = o(g(n))$ if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$. That is, for every $c > 0$, asymptotically, $f(n) < c \cdot g(n)$, (i.e., f is an order lower than g).

Problem 1. Order Notation and Recurrence Equations (30)

(a) State whether each of the following statement is true or false, with a brief reason. (i) $n^2 = O(2^n)$, (ii) $n^2 = \Omega(n^3)$, (iii) $2^{2n} = \Theta(2^{n+c})$, for any constant c, (iv) $n^2 = o(n^2 \log n)$, (v) $\log_2 n = \Theta(\log_c n)$, for any fixed constant c. (10)

(b) Solve the following recurrence equations. Assume that $T(1) = \Theta(1)$ and $T(2) = \Theta(1)$ for all the functions below. Make and state appropriate simplifying assumptions if needed. ($4 \times 5 = 20$).

1. $T(n) = T(2n/3) + n$.
2. $T(n) = 4T(n/3) + n^2$.
3. $T(n) = 3T(n/3) + n$.
4. $T(n) = 2T(n/2) + n \log n$. For this problem argue an upper bound that is $o(n^2)$.

Problem 2. Non-dominated points (35)

A two-dimensional point (x, y) is said to dominate another two-dimensional point (u, v) if $x \geq u$ and $y \geq v$. Given a set P of points, a point $p = (x, y)$ is said to be a non-dominated point of P (also called a maximal point) if no other point q in P dominates p. See Figure 1 for examples. Given a set of n points P placed in arbitrary order in an array, give a time efficient algorithm $\textsc{find\_non\_dom}(P, n)$ to find the set of all non-dominated points in P. *Notes*:

1. For simplicity, assume that no two points have the same x-coordinate or the same y-coordinate.
2. A point p is represented as a structure with two attributes $p.x$ and $p.y$. The set of points P, is represented as an array $P[1, \ldots, n]$ of points in arbitrary order.
3. You can find the index of the median of the points in $P[k, \ldots, l]$ by the x coordinate by using an informal statement like "let $i = \textsc{median}(P, k, l)$ by x-coordinate" Similarly, a statement like "let $i = \textsc{median}(P, k, l)$ by y coordinate" can be used to find the index of the median of the points in $P[k, \ldots, l]$ by the y coordinates. These functions run in time $O(k - l + 1)$. The median of n points is returned as the $\lfloor (n + 1)/2 \rfloor$ th ranked item.
4. Full marks will be provided for a correct solution that takes $O(n \log n)$ time.
5. A correct solution of $O(n^2)$ time will earn only 10 points in total.

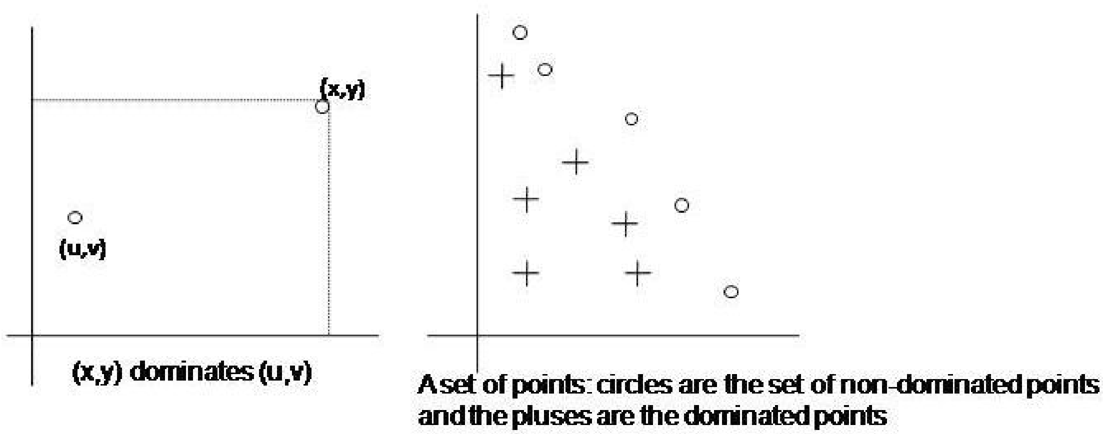

Figure 1: Dominated and non-dominated points in a point-set

Problem 3. (35)

An important court trial is going on that has N witnesses. However, not all witnesses are honest; a witness is either *true* or *false*. A *true* witness always speaks the truth, whereas a *false* witness may lie and cannot be relied upon. To test the witnesses' credibility, the judge speaks to them in pairs (e.g., $W1$ and $W2$). Both are asked the same questions about the case and the answers given by each one is presented to the other. Each of them (e.g., Wi) is now asked whether the other person (i.e. $W2$) is telling the truth or not (and vice-versa). The possibilities are as follows:

W1 says	W2 says	Conclusion
W2 is true	W1 is true	Both are true, or both are false witnesses
W2 is true	W1 is false	At least one is a false witness
W2 is false	W1 is true	At least one is a false witness
W2 is false	W1 is a false	At least one is a false witness

a. Show that if there are more than $N/2$ false witnesses, the judge cannot necessarily pick out the true witnesses, irrespective of the pairwise test strategy used. Assume that the false witnesses can conspire to fool the judge.

b. Assuming that more than $N/2$ witnesses are true, the judge now wants to identify one true witness. Show that $\lfloor N/2 \rfloor$ pairwise tests are sufficient to reduce this problem to that of approximately half its size.

c. Show that the true witnesses can be identified with $\Theta(N)$ pairwise tests, assuming that more than $N/2$ witnesses are true. Argue and solve the recurrence that describes the number of tests.