1. Consider a Monte Carlo algorithm A for a problem P whose expected running time is at most T(n) on any instance of size n and that produces a correct solution with probability f(n). Suppose further that given a solution to P, we can verify its correctness in time t(n). Show how to obtain a Las Vegas Algorithm that always gives a correct answer to P and runs in expected time at most (T(n) + t(n))/f(n).

2. Liz conjectures that for any n (≥2), n-1 is its own inverse mod n. Prove or disprove this conjecture.

3. Show the execution of ExtendedEuclidGCD(412,113) by constructing a table similar to Table 10.10 (page 465).

4. Compute successive powers for the elements of Z17 in a table similar to that of Table 10.5 (page 460).

5. Compute the multiplicative inverses of 113, 114, and 127 in Z299.

6. (Chapter 10) Alice and Bob enjoy weekly monopoly games every Friday night. When Bob is sent to the wilds of western Slobovia while Alice stays in central city, they agree that they'll just call each other up on the telephone and conduct their games that way. But they run into some interesting problems: How exactly does draw a card in a phone call, so that both know that the card draw was fair? This is what they came up with: Each card would be given a number 0 through X-1 (where X is the number of cards). Alice would pick a number and tell Bob what it was. Bob would pick a number and tell Alice what it was. The card chosen would be the sum of their numbers mod X. Will this work?

7. Draw a 2D range tree for the following points. Label them so we can tell which points you are referring to in your diagram.
8. (C-12.6) Show how to extend the two-dimensional range tree so as to answer d-dimensional range searching queries in $O(\log^d n)$ time for a set of d-dimensional points, where d is constant.

9. (C-12.4) Design a static (no insertions/deletions after construction) data structure that stores a two-dimensional set $S$ of $n$ points that can answer queries of the form $\text{CountAllInRange}(x_1,x_2,y_1,y_2)$ in $O(\log^2 n)$ time which return the number of points in $S$ with x-coordinate in the range $[x_1,x_2]$ and y-coordinates in the range $[y_1,y_2]$. What is the space used by this structure?

10. (C12.2) Give a pseudo-code description of an algorithm for constructing a range tree from a set of $n$ points (having x and y coordinates) in $O(n \log n)$ time.

11. (C-12.12) Design an $O(n)$ algorithm for testing whether a given $n$-vertex polygon is convex. $P$ may overlap itself.

12. Design an algorithm to determine whether $n$ given points are all on one line. What is the complexity of your algorithm?

13. For any set of points, is there a single, unique simple polygon whose vertices are those points? Illustrate your answer.

14. Given a set of points, $S$, and a line, design a linear time algorithm to find a line that is parallel to the given line and that separates the set of points into two equal-sized subsets. You may not assume any particular ordering of the vertices. [Hint, take a look at the randomized Quickselect algorithm.]