1. There are $\sqrt{n}$ copies of an element in the array $c$. Every other element of $c$ occurs exactly once. If the algorithm RepeatedElement is used to identify the repeated element of $c$, will the run time still be $\tilde{O}(\log n)$? If so, why? If not, what is the new run time?

2. Let $A$ be a Monte Carlo algorithm that solves a decision problem $\pi$ in time $T$. The output of $A$ is correct with probability $c$, $c$ being a constant greater than $1/2$. Show how you can modify $A$ so that its answer is correct with high probability. The modified version can take $O(T \log n)$ time.

3. Input are two $n \times n$ matrices $A$ and $B$. The problem is to check if $A = B$. It is known that if $A \neq B$, then these two matrices will differ in at least $n$ elements. Present a Monte Carlo algorithm for this problem that runs in $O(n \log n)$ time.

4. [Problem 7.2 from MR95.] Two rooted trees $T_1$ and $T_2$ are said to be isomorphic if there exists a one-to-one mapping $f$ from the vertices of $T_1$ to those of $T_2$ satisfying the following condition: for each internal vertex $v$ of $T_1$ with the children $v_1, \ldots, v_k$, the vertex $f(v)$ has as children exactly the vertices $f(v_1), f(v_2), \ldots, f(v_k)$. Observe that no ordering is assumed on the children of any internal vertex. Devise an efficient randomized algorithm for testing the isomorphism of rooted trees and analyze the performance.

5. [Problem 7.10 from MR95]. Given a randomized algorithm for testing the existence of a perfect matching in a graph $G$, describe how you would actually construct such a matching. What is the run time of your algorithm if you use the testing algorithm described in class?

6. [Problem 7.13 from MR95]. Consider the two-dimensional version of the pattern matching problem. The text is an $n \times n$ Boolean matrix $X$ and the pattern is an $m \times m$ Boolean matrix $Y$. A pattern match occurs if $Y$ appears as a ( contiguous) sub-matrix of $X$. To apply the randomized algorithm described in class, we can convert $Y$ into an $m^2$-bit vector using the row-major format. The possible occurrences of $Y$ in $X$ are the $m^2$-bit vectors $X(j)$ obtained by taking all $(n - m + 1)^2$ sub-matrices of $X$ in a row-major form. It is clear that the algorithm discussed in class can be used in this case. Analyze the error probability and run time in this case.