1. Explain how Quicksort can be made to run in $O(n \log n)$ time in the worst case.

2. Let $S$ be a multiset of $n$ integers (i.e., elements can repeat in $S$). Give an algorithm running in $O(n)$ time and space for determining whether an element occurs precisely $\lceil n/2 \rceil + 6$ times in $S$.

3. A bit vector is an array bits (0s and 1s). A bit vector of length $m$ takes much less space than an array of $m$ pointers. Describe how to use a bit vector to represent a dynamic set of distinct elements with no satellite data. Dictionary operations should run in $O(1)$ time.

4. Demonstrate what happens when we insert the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be $h(k) = k \mod 9$.

5. Let $S$ be a multiset of $n$ integers (i.e., elements can repeat in $S$). The goal is to find two elements $s, s' \in S$ such that $s + s' = 0$.

(a) How many solutions can there be (in terms of $n$)?
(b) Explain why any algorithm for the problem runs in $\Omega(n)$ time.
(c) How fast of an algorithm, in terms of worst-case performance, can you devise for the problem with hashing? What about without hashing?