Q1. Suppose that we use an open-addressed hash table of size m to store $n = \alpha m$ items, where $\alpha < 1$. Assume uniform hashing. Let the random variable X_i denote the number of probes required by the ith insertion, and $X = \max_{1 \le i \le n} X_i$ be the maximum number of probes required by any of the n insertions.

1. **(1 point).** Show that for $i = 1, 2, \ldots, n$, the probability is $O(1/n^2)$ that the ith insertion requires strictly more than $2 \log_{\frac{1}{\alpha}} n$ probes.

2. **(2 point).** Show that $Pr[X > 2 \log_{\frac{1}{\alpha}} n] = O(1/n)$ (*hint:* use the *union bound* $Pr[A \cup B] \le Pr[A] + Pr[B]$).

3. **(2 point).** Show that the expected length $E[X]$ of the longest probe sequence is $O(\log_{\frac{1}{\alpha}} n)$.

Q2. There were two quizzes and n students. All students attended the first quiz, and all but one attended the second. The teacher kept a single list containing, not in any particular order, the ID's of the students attending the quizzes (where the ID of a student appears twice if (s)he attends both quizzes). The teacher would like to find the ID of the student who was absent from the second quiz.

Design three algorithms to solve this problem, with the following complexities, and explain why they have these complexities:

1. **(1 point)** *Algorithm 1*: runs in $\Theta(n \log n)$ time and uses $O(1)$ additional space;

2. **(2 points)** *Algorithm 2*: runs in $O(n)$ time and uses $\Theta(n)$ additional space;

3. **(2 points)** *Algorithm 3*: runs in $O(n)$ time uses $O(1)$ additional space.