Problem 1: (10 points) In the first lecture we analyzed the following counter algorithm which counted up to \( n \) using much less than \( O(\log n) \) bits of space: initialize a counter \( X \) to 1, and for every increment instruction, increment \( X \) with probability \( \frac{1}{2^X} \). By averaging many such estimators, we obtained a \( (1+\varepsilon) \)-approximation to \( n \) with good probability. Here we will investigate a different way to obtain a good approximation. Imagine we still initialize \( X \) to 1, but we increment it with probability \( \frac{1}{1+a} \) instead. (Note: our estimator for \( n \) would have to change from \( 2^{X-1} \) to something else; figure out what!)

**Question:** How small must \( a \) be so that our estimate \( \hat{n} \) of \( n \) satisfies \( |n - \hat{n}| \leq \varepsilon n \) with at least \( \frac{9}{10} \) probability when we return the output of a single estimator instead of averaging many estimators as in class? Also derive a bound \( S = S(n) \) on the space (in bits) so that this algorithm uses at most \( S \) space with at least \( \frac{9}{10} \) probability by the end of the \( n \) increments.

Problem 2: (10 points) In class we saw that there is no deterministic algorithm using \( o(n) \) bits of space which approximates the number of distinct elements in a stream up to a factor 2, where the stream tokens come from the universe \( \{1, \ldots, n\} \). Show that there is no randomized algorithm for the distinct elements problem using \( o(n) \) bits of space which solves the problem exactly with at least \( \frac{99}{100} \) probability on any input. Thus, any \( o(n) \) space algorithm must be both randomized and approximate.

Problem 3: Recall the AMS sketch from class for \( F_2 \) moment estimation: a random \( m \times n \) matrix \( \Pi \) with entries \( \pm \frac{1}{\sqrt{m}} \) is drawn for \( m = O(1/\varepsilon^2) \), and \( \|x\|_2^2 \) is estimated as \( \|\Pi x\|_2^2 \). Then with at least 2/3 probability,

\[
(1 - \varepsilon)\|x\|_2^2 \leq \|\Pi x\|_2^2 \leq (1 + \varepsilon)\|x\|_2^2. 
\] (1)

(a) (5 points) Imagine picking \( \Pi \) differently: for each \( i \in \{1, \ldots, n\} \) we pick a uniformly random number \( h_i \in \{1, \ldots, m\} \). We then set \( \Pi_{h_i,i} = \pm 1 \) for each \( i \in \{1, \ldots, n\} \) (the sign is chosen uniformly at random from \( \{-1,1\} \)), and all other entries of \( \Pi \) are set to 0. This \( \Pi \) has the advantage that in turnstile streams, we can process updates in constant time. Show that using this \( \Pi \) still satisfies the conditions of Equation 1 with 2/3 probability for \( m = O(1/\varepsilon^2) \).

(a) (5 points) Show that the matrix \( \Pi \) from Problem 3(a) can be specified using \( O(\log n) \) bits such that Equation 1 still holds with at least 2/3 probability, and so that given any \( i \in \{1, \ldots, n\} \), \( \Pi_{h_i,i} \) and \( h_i \) can both be calculated in constant time.