1. In lecture, we saw how to choose hash functions that satisfy a stronger property than universality, **pairwise independence**: that is, for a hash function $h_a$ chosen uniformly at random from the family $\mathcal{H}$, for any two elements $x$ and $y$ in the universe and any two values $w$ and $z$ in the range $(0, \ldots, m-1)$, $Pr_a[h_a(x) = w \text{ and } h_a(y) = z] = 1/m^2$.

Here is an application of pairwise independent hash functions in message authentication: suppose that Alice and Bob secretly agree on a member of $h_a \in \mathcal{H}$. Then, when Alice wishes to send a message $x$ to Bob, she sends $x$ attached to $w = h_a(x)$. Bob then checks that the pair $(x, w)$ he receives satisfies $w = h_a(x)$, and if so, accepts $x$ as genuine. (Otherwise he rejects the message.)

Show that for a suitable choice of $m$, no adversary who intercepts $(x, w)$ can corrupt it to some $(y, z)$ for $y \neq x$ that Bob will accept as genuine except with probability at most $\delta$, even if the adversary knows what family $\mathcal{H}$ Alice and Bob are using and can spend unlimited computational resources crafting $(y, z)$ from $(x, w)$. How large does $m$ (and hence, the tag $w$) need to be?

2. Recall the algorithm **insertion sort**:

   ```
   input : Array $A$ of $n$ integers
   begin
   for $i = 1, \ldots, n$ do
     for $j = i, \ldots, 1$ until $A[j-1] \leq A[j]$ do
     end
   end
   end
   ```

   Suppose $A$ contains $n$ integers in a uniformly chosen random order. What is the average (expected) running time of insertion sort?

3. **Kleinberg & Tardos** Chapter 8, question 32