10. tutorial

Binary counter. We implement a binary counter in the straightforward way: a number n is represented by $\lceil \log n \rceil$ bits, and the i-th bit is 1 if it appears in the binary representation of n. You can think of the bits as an array.
The `INC` operation increments the counter – after it is finished, the counter should represent $n+1$. Thus, it iterates over the bits starting from the least significant one, and as long as the bits are "1", it flips them to 0, and it finally flips a "0" to "1".
Prove that `INC` has amortized complexity $\mathcal{O}(1)$.
Hint: Use the charging method: charge every `INC` operation a certain fixed amount of coins (say, 4) and say that each elementary task (e.g., flipping a bit) costs one coin. If an operation doesn't use all of its coins, store them in the bank. Then, you must show that each operation will have enough coins to pay for the tasks it needs to perform.

Counter with `DEC`. Again consider an n-bit counter, but now it should support both an `INC` and a `DEC` operation (increment / decrement, respectively), and also a `TESTZERO` operation which reports whether the represented number is 0.
How would you implement this counter, so that the complexity of `INC` and `DEC` is amortized $\mathcal{O}(1)$, and the complexity of `TESTZERO` is worst-case $\mathcal{O}(1)$?
There are at least 2 ways to go about it. One is to allow each bit to attain values 0, $+1$ and -1. The other is to get two "regular" binary counters, but maintain a certain invariant about them (which requires modification of the `INC` operation).
In both cases, one can show the amortized cost by the charging method.

Queue. Show how to use two stacks to simulate a queue. Try to have $\mathcal{O}(1)$ amortized complexity for the operations (assuming that the stack works in constant time); use the charging method as before.

Collision. You were given a hash function $h : [U] \to [m]$. Unless you know anything else about the function, how many function evaluations do you need to find a set of k elements of $[U]$ which all collide, that is, $\{a_1, \dots, a_k\} \subseteq [U]$ and $h(a_1) = \cdots = h(a_k)$?