Problem Exercise 22.1**.** Recall the array-backed lists from Section 22.3, where we showed how to support `append` in $O(1)$ amortized time via the "doubling trick". We would like to support a `remove` operation with similar amortized time. Consider the following approach:

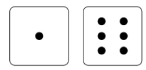
> After removing the element from the array and decreasing the size counter, if the size of the list is now less than half of the capacity of the array, allocate a new array of half the capacity and copy the elements of the list over.

1. Show that this does not achieve $O(1)$ amortized time. That is, give a sequence of n `append`/`remove` operations on which the data structure would take $\Omega(n^2)$ time.
2. While the approach above didn't quite work, a slight modification of it will. Implement remove so that append and remove both take O(1) amortized time

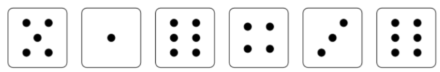
Problem Demonstrating mass. Prove or disprove: for any event A, B:

$$\Pr[A \wedge B] \leq \min\{\Pr[A], \Pr[B]\}.$$

Exercise 25.9. Recall that when we roll six-sided dice, the dice samples an integer between 1 and 6 uniformly at random. Let us call an unordered pair of dice "lucky" if one of them is a 1 and the other is a 6.



If we roll 6 independent six-sided dice, how many lucky pairs do we expect? Note that a single dice may appear in more than one lucky pair. For example, the following roll of six dice has 2 lucky pairs amongst them.



Sometimes you just need to count.

Exercise 25.10. Suppose you repeatedly flip a coin that is heads with fixed probability $p \in (0, 1)$.

1. What is the expected number of coin flips until you obtain one heads?[4] Prove your answer.
2. What is the expected number of coin flips until you obtain two heads? Prove your answer.
3. For general $k \in \mathbb{N}$, what is the expected number of coin tosses until you obtain k heads? Prove your answer.

Expectation