### Calculating total games played

If we want to calculate the total number of games played, we need to simplify the game and define all of its parts. For this first part, let’s assume a simple game that consists of town players and mafia players where . The game will end either if the number of mafia players outnumber the town players () or if the number of mafia players is equal to 0 (). If the game doesn’t end, then it will enter a night/day cycle. During the day, 3 things can happen:

1. The town successfully kills a mafia
2. The town kills another town
3. The town doesn’t kill anyone

For the night, let’s assume that the mafia kills a town every time.

Notice that after each night/day cycle, the same game is run, except with fewer players. This means that we can build a recursive function to count all of the possible games. Let’s call it. Our base case happens when the game ends, which we already determined earlier on. Out function should return 1 in this case.

Otherwise, we can call our function again for every possible outcome of the night/day cycle and add them together to get the total number of games played. Outcome 1 gives us, output 2 gives and output 3 gives. We always subtract 1 from the number of town since the mafia is killing a town every night. Putting this all together gives us the following function:

If we want to see how many times mafia wins, we only need to make one small adjustment to our function. Instead of returning 1 at the end of the game, we return 1 only if the game ends in a mafia win. If the town wins, we want to return 0 instead.

We can divide this value by the total number of games to estimate the likelihood that the mafia will win a given game. For example:

Looking at this function makes it clear that the town deciding not to kill anyone is always a bad idea in this simplified version of the game. Removing that option from the function lowers the mafia win rate and helps simplify the function a bit.

### Predicting game outcomes

If we want to be more accurate with our mafia win rate prediction, we need to take the probabilities of certain events into account, as not every outcome has an equal likelihood of occurring. First, let’s build a function that predicts the outcome of a simple game. If the town wins it will return 0, and if the mafia wins it will return 1. We can use the same base cases as our number of mafia wins function.

Next, instead of calling our function again for all possible outcomes, we want to call it once, depending on what happens during the day. If a town dies, we will call and if a mafia dies, we will call. We can predict what happens during the day by choosing a random player, and checking to see whether or not they are mafia. That can be accomplished by the following steps:

1. Choose a random number from 0 to (total number of players)
2. If < then you have chosen a mafia player
3. Otherwise, you have chosen a town player

Based on what player we choose, call the corresponding function.

This function predicts whether or not mafia wins for a single game, so running this function over many games should give a more accurate estimate for mafia win rates. For example, running the function 1,000,000 times results in:

This is quite a high win rate for the mafia; let’s add something to our prediction to help the town. In the game of mafia, there are several special roles that the town can take on, one of them being the detective (or cop).