538 Riddler Express

Evan Rittner

9 April 2021

Problem

Three of Matt's students — Players A, B and C — are engaged in a game of *veinte*. In each round, players take turns saying numbers in order (Player A, then B, then C, then A again, etc.). The first player to go says the number "1." Each number must be either one, two, three or four more than the number said by the previous player. When someone says "20," the round is over and the *next* person is eliminated, with the following person beginning the subsequent round. For example, if Player A says "20," then Player B is eliminated, while Player C begins the next round by saying "1." At no point can anyone say a number greater than 20.

All three players want to be the winner (i.e., the only player remaining) after the two rounds. But if they realize they can't win, then they will prioritize making it to the second round. Player A starts things off by saying "1." Which player will win?

Extra credit: Instead of three players, now suppose there are four — Players A, B, C and D — all of whom want to make it through as many rounds of the game as possible. Again, Player A starts things off by saying "1." Which player will win?

Solution

In the three player game, Player B wins (and Player A advances to the next round). In the extra credit four player game, Player C wins, with B advancing to the round of two, and D advancing to the round of three.

To solve this problem, I'll determine an algorithm from simple cases, then generalize it and let a computer do the heavy lifting. Let's consider strategies at the end of the "round of two", when only two players remain. In the left column is the number that the previous player just said. In the center column is the optimal strategy for you to say in response, and in the right column is your final outcome following the optimal strategy. If your opponent says "16", "17", "18", or "19", the optimal strategy is simple: you can just say "20" and win the game (Fig. 1).

If your opponent says a number smaller than 16, it is less obvious the ideal move, since you can say a number no more than four larger than your opponent's, and thus cannot just end the game. We can consider between the four options (saying one more, two more, three more, or four more) by using the table in Fig.

Opponent's Move	Optimal Response	Outcome
19	20	Win
18	20	Win
17	20	Win
16	20	Win

Figure 1: Optimal moves for numbers 16 - 19 in the round of two

1 from the opposite perspective: if you say "16", your opponent's optimal strategy is to say "20" and win; therefore, your outcome of saying "16" results in your loss. In this case, all four options result in your loss. We'll say arbitrarily that the optimal strategy in this case is to say "19", the largest of the options. We can now finish the table for the round of two:

Opponent's Move	Optimal Response	Outcome
19	20	Win
18	20	Win
17	20	Win
16	20	Win
15	19	Lose
14	15	Win
13	15	Win
12	15	Win
11	15	Win
10	14	Lose
9	10	Win
8	10	Win
7	10	Win
6	10	Win
5	9	Lose
4	5	Win
3	5	Win
2	5	Win
1	5	Win

Figure 2: Optimal moves for the round of two

Since the first player must always start by saying "1", we can see the second player's final outcome in the last row of the table: the optimal response to the opponent saying "1". In the round of two, the second player will win (and the first player will lose).

Continuing this strategy for larger rounds is quite similar, though with a few complicating factors. With larger rounds, it is no longer a binary win/loss, but a range of possible final outcomes. For example, in the round of three, there are three outcomes. From best to worst, they are 1) Avoiding elimination in every round, and thus winning the whole game; 2) Advancing to the round of two, but being eliminated there; and 3) Being eliminated in the round of three.

With this greater number of outcomes, the desired end state of the round is no longer obvious. In the round of two, the winner is the player that says "20", but what is the best ending state in the round of three?

Well, we know that the second player in the round of two has the most desirable final outcome (winning the whole game), followed by the player going first, followed by the player eliminated in the round of three. Therefore, the best finishing position in the round of three is to be the player that says “20”: the player one after is eliminated, and the player two after starts the round of two, going first.

We are almost ready to generalize this algorithm, but there is a final consideration to make. In the round of two, we looked at what outcome we would force our opponent into by making a particular move. If our move would force them to eventually lose, then we would eventually win. However, with more outcomes, this correspondence is less clear. It can be determined by examining the end state. For example, in the round of three, if we force the next player to win (i.e. to begin the round of two playing second), what is our eventual outcome? If they win, they were the player that said “20”, and since we are one spot ahead of them (equivalently, two behind them), we are advanced to the round of two, though in the losing position.

Following this algorithm through yields the winning positions in the rounds of three and four, being Players B and C, respectively.

Finally, I’d like to show some visualizations of the game for much larger sizes. Below are a series of images, of different levels of zoom, of a 2D plot. Each row represents a round, from the round of one, increasing each row. The brightness of each pixel in each row represents the final outcome of a player starting in that position. For example, in the round of four, we have determined that the first player will lose in the fourth round, so the first pixel in the fourth row is black. The second player will advance to the final round before being eliminated, so the second pixel is a light gray. The third player will win the entire game, so that pixel is white.

Here is a link to a Google Drive folder with a spreadsheet that shows the logic for the rounds of three and four, and two Python files: one calculates the final outcomes for each starting position, given a particular round, and the other produces the following visualizations (which are also in the shared folder).

`https://drive.google.com/drive/folders/1lIFgOal9NkzDFmH_kptYa6AOQusmO87T?usp=sharing`

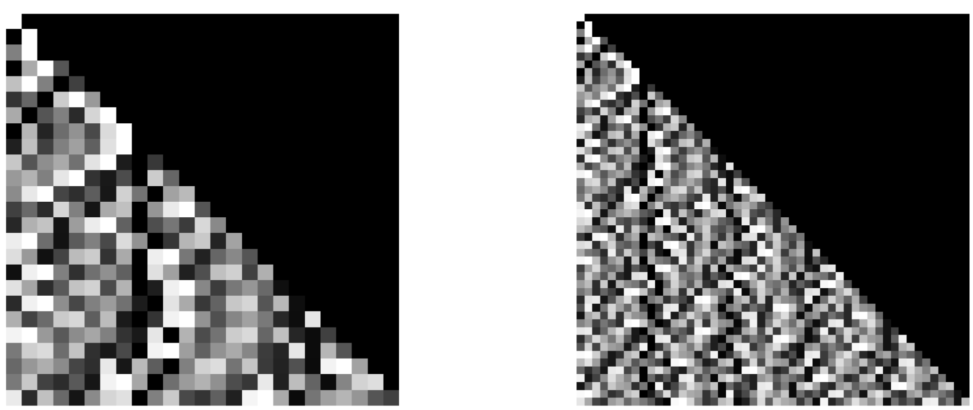

Figure 3: Visualization to $n = 25$ and $n = 50$

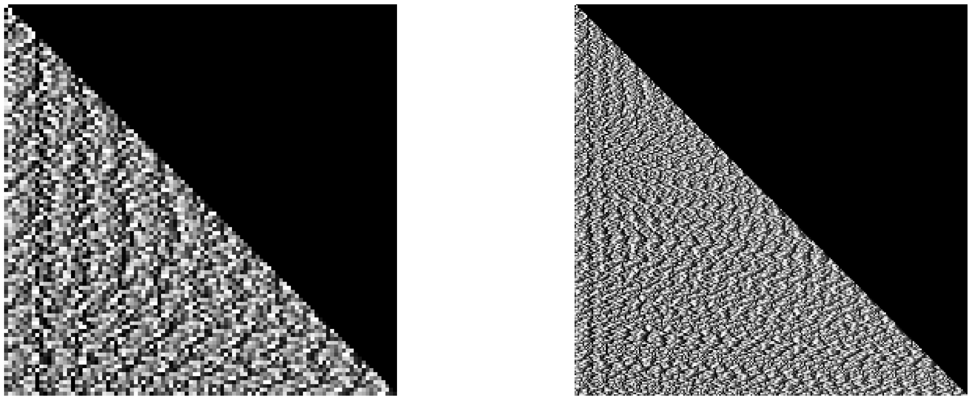

Figure 4: Visualization to $n = 100$ and $n = 250$

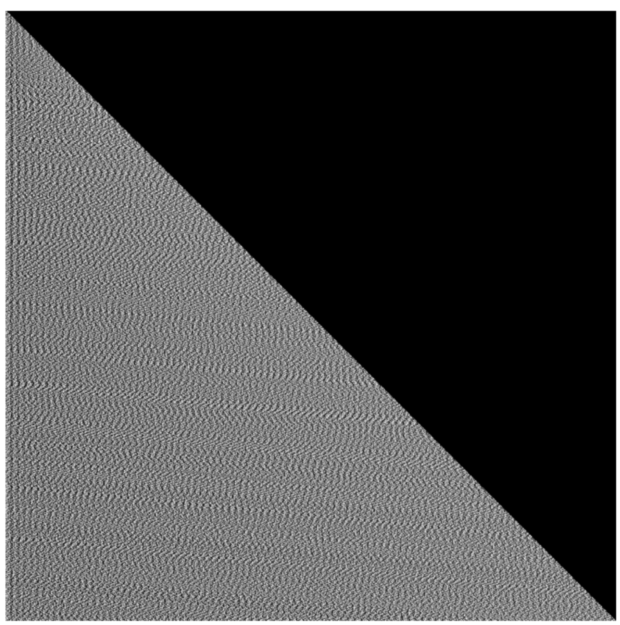

Figure 5: Visualization to $n = 1000$