Catch-Up: A Game in Which the Lead Alternates

Aaron Isaksen Tandon School of Engineering, New York University

Mehmet Ismail Department of Economics, Maastricht University

Steven J. Brams Department of Politics, New York University

Andy Nealen Tandon School of Engineering, New York University

Catch-Up is a two-player game in which the players' scores remain close throughout the game, making the eventual winner – if there is one – hard to predict. Because neither player can build up an insurmountable lead, its play creates tension and drama, even between players of different skill. We show how the game is played, demonstrate that its simple rules lead to complex game dynamics, analyse some of its most important properties, and discuss possible extensions.

1 Introduction

It is a challenge to design interesting two-player games with simple rules that keep the score close, even between players of different skill. When the game score is close, players experience *tension* and *drama* by not knowing too far in advance who will win. This drama has been discussed qualitatively [1, 2] and quantitatively [3].

To enhance tension, games often have catch-up mechanisms, sometimes called *rubber banding* [1]. Players who are behind can receive a boost to help them recover, and players who are ahead are prevented from maintaining or accelerating their lead.

Economists describe the desire to minimise inequality as *inequity aversion*, wherein people prefer rewards to be allocated evenly [4]. Designing games with inequity aversion can create a more balanced competitive experience, allowing experts and novices to enjoy playing together as the score will remain close throughout the game. A game is also often more enjoyable if one is not losing by a large amount. However, too much catching up can lead to games in which the winner is not determined until the very end, making early moves meaningless.

1.1 Catch-Up

We present Catch-Up, a *minimal game* [5] with simple rules that can be learnt quickly, invented by the authors with these ideas in mind.[1] The rules are as follows.

Catch-Up starts with a set of numbers S.

1. Two players, P_1 and P_2, begin with scores, s_1 and s_2, of zero. P_1 starts by removing a number from S, which is added to his or her score.

2. The players then take turns removing one or more numbers from S, one by one, until the acting player's score equals or exceeds the opponent's current score.

3. When this is no longer possible, the acting player receives any remaining numbers. The player with the higher score wins; the game is drawn if scores are tied.

Catch-Up provides meaningful choices, with score balancing built into its rules. Players alternate holding the lead, with the score difference bounded by a relatively small number. Note that the game is actually played with a *multiset* of numbers, i.e. some numbers may be repeated, but we use the term 'set' here for simplicity.

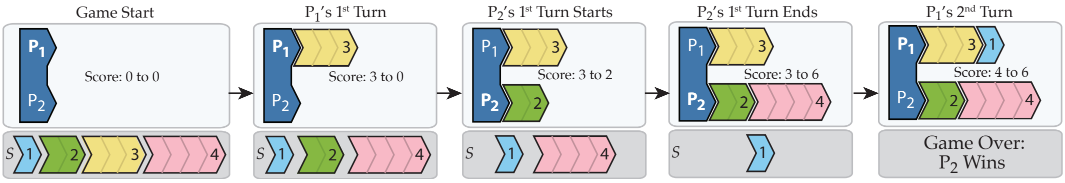


Figure 1. An example game of Catch-Up won by player P_2.

[1] The demo game and code used in this paper are available at: http://game.engineering.nyu.edu/catch-up

Isaksen, A., *et al.*, 'Catch-Up: A Game in Which the Lead Alternates', *G&PD*, vol. 1, no. 2, 2015, pp. 38–49. © 2015

Players are therefore uncertain who will win a game of Catch-Up until the end. The game is surprisingly complex, given the simplicity of its rules, with no trivial heuristics that enable players to win every time.

We illustrate with several examples how Catch-Up is played, discuss optimal strategies and heuristics, analyse some important properties, and discuss possible extensions of the game. Figure 1 shows a short game won by player P_2, by way of example.

1.2 Combinatorial Aspects

We study this combinatorial game using an approach similar to that used in *Scientific American* articles by Martin Gardner [6] and *Winning Ways For Your Mathematical Plays* [7], but we also provide context for game designers. In particular, we show how the set – the numbers the players start with – affects the game's complexity and play dynamics.

Catch-Up is a combinatorial perfect information game, so even though the players have close scores throughout the game, there exist optimal strategies to win or tie. Thus, the scoring mechanism does not necessarily reflect who is more likely to win the game: a player may be in a game-theoretic winning position even though his or her score is lower than the other player's. Catch-Up shows that creating scoring systems in which the current score is a reliable and meaningful indicator, in games with significant catch-up mechanisms, is indeed a challenge.

Catch-up mechanisms exist in many games, from board games using variable scoring (e.g. Hare & Tortoise [8]) or time tracks (e.g. Tokaido [9]) to video games with variable powerups (e.g. Super Mario Kart [10]). The game Catchup, by Nick Bentley,[2] uses a catch-up rule that permits the player who is behind to add an extra piece each turn. Zhang-Qi[3] is similar (though we were not aware of it when designing ours) but uses a specific 32-element set, places markers on a uniquely shaped board, and describes the catch-up rule as one of two optional rules.

2 Examples of Play

The rules of Catch-Up are presented in the shaded box on the preceding page. We explore several example games to show that the rules, although minimal, define a game with interesting non-trivial properties.

We use the notation Catch-Up(S) to describe the game played with set S. For example, Catch-Up($\{1, ..., N\}$) is played with $S = \{1, ..., N\}$, the consecutive positive integers from 1 through N. For clarity, P_1 is referred to as she and P_2 as he.

2.1 Catch-Up($\{1, ..., 4\}$)

Figure 2 shows the full game tree for Catch-Up($\{1, ..., 4\}$). Assuming optimal play by P_1 (triangle) and P_2 (square), winning, drawing, and losing positions, and moves for the acting player are indicated. Numbers show the numbered pieces selected on that move. Thicker lines indicate optimal plays. Above each node, $\{B\}$:D gives the remaining numbers B and the score differential D.

One possible game might play out as follows, which is shown in steps in Figure 1. The set starts with $S = \{1, 2, 3, 4\}$. P_1 initially removes $\langle 3 \rangle$, and is ahead $3 - 0$. Play then switches to P_2, who can choose from $\{1, 2, 4\}$ and removes $\langle 2 \rangle$. Since the score is $3 - 2$ and P_2 is still behind, P_2 needs to remove another number. P_2, choosing from $\{1, 4\}$, removes $\langle 4 \rangle$. Thus, on P_2's turn the entire move was to remove $\langle 2, 4 \rangle$, and the score is now $3 - 6$. Since P_2 is ahead, play switches back to P_1.

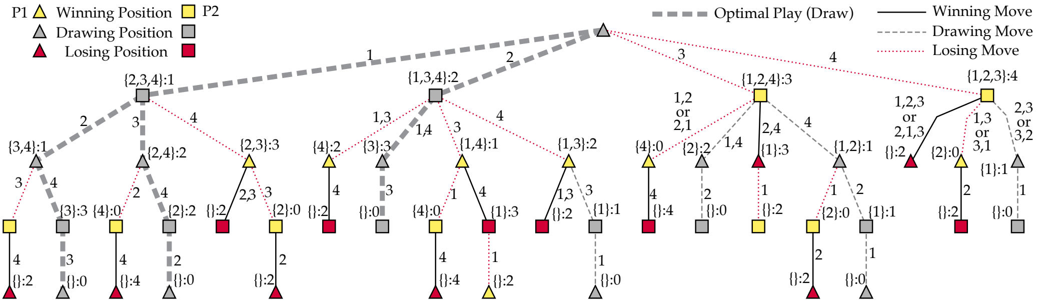


Figure 2. The full game tree for Catch-Up({1,2,3,4}).

[2] https://nickbentleygames.wordpress.com/2012/04/29/my-best-game-i-suspect-ketchup/

[3] http://boardgamegeek.com/boardgame/72711/zhang-qi

The set contains only $\{1\}$, which P_1 removes. The game ends with a final score of $4-6$, so P_2 is the winner by 2. This game could also have ended in a draw as follows: P_1 selects $\langle 2\rangle$, P_2 selects $\langle 1,4\rangle$, and P_1 selects $\langle 3\rangle$, tying the game at $5-5$ and illustrating how P_2 can force a draw.

Because of Rule 2, players always start their turns either tied or behind the other player. This means the player's task is at least to catch up to the other player, but neither player can 'snowball' or jump far ahead. Conversely, this same rule means that players will always end their turns either tied or ahead of the other player.

In order to keep players from memorising strong opening moves, we propose that players play with a randomised set – with repeated or missing numbers – such that there are too many possible game trees for players to memorise.

2.2 Physical Implementation

If Catch-Up is played as an abstract mathematical game, it requires detailed bookkeeping, which some players may find difficult. We propose a version played with physical pieces on a table, as shown in Figure 1, illustrating the moves described in Section 2.1. The pieces are designed to fit next to each other, such that that the lengths can be quickly determined to see whose turn it is.

We believe the physical version is more pleasurable to play because the physical pieces simplify the arithmetic calculations, making the game more accessible [11]. If the shortest piece is 1 centimetre long, a tie game of Catch-Up($\{1,...,12\}$) would end up being $(1+...+12)/2 = 39$ cm long, with the largest win margin at most 12 cm long.

2.3 Puzzle-Like Quality

Catch-Up has a puzzle-like quality [3], making it challenging to find solutions that lead to a win or draw. For example, Figure 3 shows two subtrees of Catch-Up($\{1,...,7\}$).

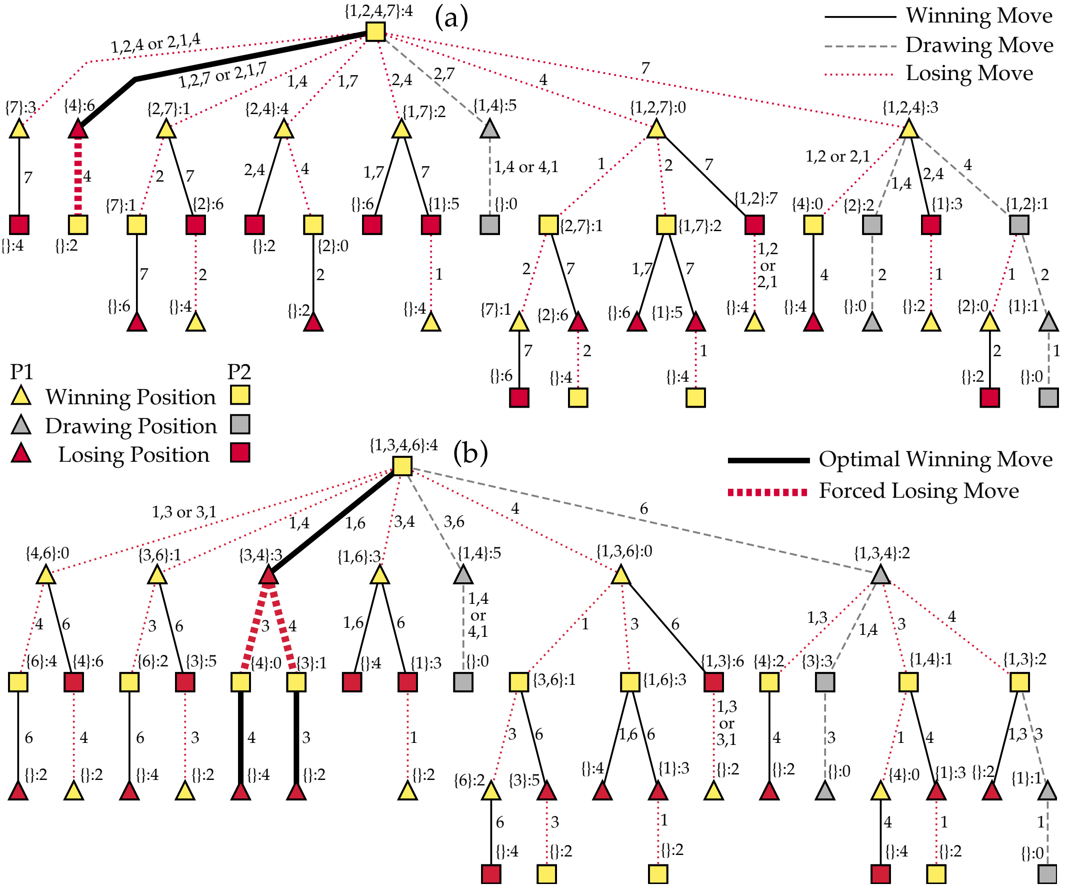


Figure 3. Two subtrees of Catch-Up($\{1,...,7\}$).

In Figure 3a (top tree), P_2 (represented by squares) is in a winning position, but he must proceed carefully. This position was reached by P_1 (represented by triangles) initially removing $\langle 3 \rangle$, P_2 removing $\langle 5 \rangle$, and P_1 removing $\langle 6 \rangle$, leaving a set of $\{1,2,4,7\}$ and a score difference of 4. One move leads to a win, one move leads to a draw, and all other moves lead to losses (random play here would lead to a 7/8 chance of choosing a sub-optimal move). P_1's optimal move is to choose the largest sum possible, removing either $\langle 1,2,7 \rangle$ or $\langle 2,1,7 \rangle$. Each uses the same numbers and reaches the same score; strategically equivalent moves are indicated with 'or' in Figure 3.

This strategy of maximising one's lead, however, does not always work. The subtree of Figure 3b (bottom tree) is reached by P_1 initially removing $\langle 2 \rangle$, P_2 removing $\langle 5 \rangle$, and P_1 removing $\langle 7 \rangle$, giving a score difference of 4. If P_2 then maximises his score by choosing $\langle 3,6 \rangle$, worth 9 points, this leads to a forced draw. But if P_2 chooses the lower valued $\langle 1,6 \rangle$, worth only 7 points, he forces a win. Making this even more tricky, choosing $\langle 3,4 \rangle$, also worth 7 points, leads to a forced loss for P_2.

In Section 3 we discuss various simple strategies and heuristics that beginning players might use to help navigate the game tree. This shows the relative effectiveness of each heuristic.

2.4 Maximising Is Not Optimal

In the previous section, we showed that a strategy of selecting the largest sum of numbers possible is not always an optimal strategy, though it is an obvious heuristic that a player might try. As another example, in a game of $N = 5$, with a set $\{1,2,3,4,5\}$, if P_1 always selects the numbers that gives her the largest lead, she will lose: P_1 initially removes $\langle 5 \rangle$, P_2 can then remove $\langle 1,3,4 \rangle$, forcing P_1 to choose $\langle 2 \rangle$ and lose the game $7-8$.

This happens specifically because of the inequity aversion of Rule 2. If, for example, players were required to select a fixed number of numbers on each turn; then a maximising-score strategy would be dominant, making the game trivial. By contrast, the rules of Catch-Up lead to a game tree that makes optimal choices non-trivial: there are no immediately obvious strategies to win *every* game.

2.5 Endgame

On every turn, a player of Catch-Up comes from behind or from a tied score. However, there are many cases in which a player, who will lose if the opponent plays optimally, can still come back to win very late in the game if the opponent makes a mistake on his or her last move. This implies that both players must focus on winning up until their very last moves.

In Figure 4, we show an example of a subtree of Catch-Up($\{1, ..., 7\}$) wherein optimal play produces a loss for P_1, but there is still a chance for a win with the last moves in the game if P_2 plays non-optimally. To reach this position, assume P_1 chooses $\langle 1 \rangle$, P_2 chooses $\langle 2 \rangle$, P_1 chooses $\langle 5 \rangle$, and P_2 chooses $\langle 6 \rangle$, so the score difference is 2 and $\{3,4,7\}$ remain in the set. Now, if P_1 chooses $\langle 7 \rangle$, she will lose when P_2 is forced to choose $\langle 3,4 \rangle$. However, if P_1 chooses $\langle 3 \rangle$ or $\langle 4 \rangle$ – putting her 1 or 2 ahead – then P_2 must choose $\langle 7 \rangle$ to win.

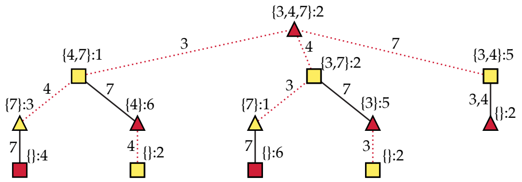


Figure 4. P_2 can force a win in this subtree.

2.6 Drawn Games

Drawn games are sometimes possible in Catch-Up if the sum of the numbers in S is even. Whether optimal play leads to a draw, or a win for P_1, depends on S. Games that end in a draw may be dissatisfying for some players because there is no winner (although draws do not seem to bother many Chess players, for example).

Whether Catch-Up permits draws is solely determined by the set S. In the case of Catch-Up($\{1, ..., N\}$), it depends on the value N. For all $n \geq 0$, games of the form $N = 4n + 1$ and $N = 4n + 2$ always have a winner by at least one point, because the sum of all the points $1, 2, ..., N$ is odd; there is no way to split them evenly. Conversely, games of the form $N = 4n + 3$ or $N = 4n + 4$ can have games that end in a draw, because the sum of all the numbers is even. We provide a proof of this in Appendix 5.1, and we calculate in Section 4.9 how often draws will occur as a function of N.

For games of the form Catch-Up($\{1, ..., N\}$) with $N = 4n + 3$ or $N = 4n + 4$, which can have games that end in a draw, we have calculated up to $N = 20$ that optimal play by both players leads to a draw (see Section 3.1). However, optimal play in any even-sum game of Catch-Up(S) for any arbitrary S does not necessarily produce a draw. Consider an even-sum game with repeated numbers $S = \{2,2,2,3,3\}$, shown in Figure 5, which sum to 12. Here P_1 can force a win by initially choosing $\langle 2 \rangle$. Drawn games are still possible for this set, but they are not the result of optimal play.

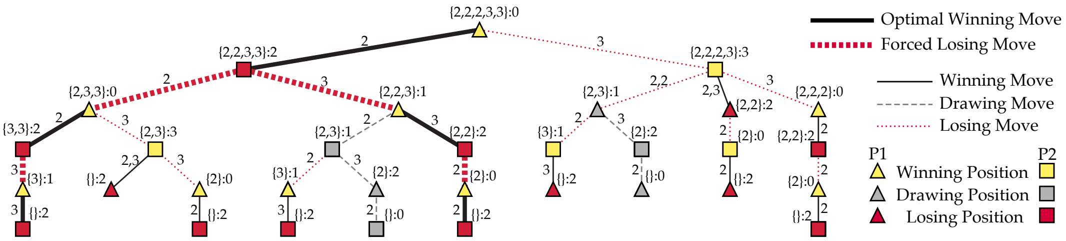


Figure 5. An even-sum game in which P_1 can force a win, but which can also end in a draw.

Furthermore, it is easy to see that some even-sum games do not even permit a draw. Consider Catch-Up($\{2, 4, 6, 8, 10\}$), which is even-sum, but obviously no subsets of these numbers can produce a $15 - 15$ tie.

2.7 Importance of the First Move

One criticism of catch-up type mechanisms is that the early moves in the game have no importance. We show here that the first move P_1 makes in Catch-Up($\{1, ..., 7\}$) has an impact on the percentage of ways that P_1 can win, lose, or draw. In Table 1, each row shows the change from 50%–50% in the percentage of ways that the game can end in a win, lose, or draw, given that P_1 makes the indicated first move.

Move	Δ Win%	Δ Lose%	Δ Draw%
$\langle 1 \rangle$	0.60%	0.60%	-1.19%
$\langle 2 \rangle$	-2.46%	3.65%	-1.19%
$\langle 3 \rangle$	6.43%	-6.90%	0.48%
$\langle 4 \rangle$	3.10%	-1.90%	-1.19%
$\langle 5 \rangle$	0.32%	-0.79%	0.48%
$\langle 6 \rangle$	-4.13%	3.65%	0.48%
$\langle 7 \rangle$	-3.85%	1.71%	2.14%

Table 1. Percentage change with P_1 moving first.

By choosing $\langle 3 \rangle$, P_1 increases the ways of winning by 6.43% and reduces the ways of losing by 6.90%. Conversely, choosing $\langle 6 \rangle$ decreases the ways of winning by 4.13% and increases the ways of losing by 3.65%. Clearly, the first move has an impact on the ability of non-optimal players to achieve a win, loss, or draw; but this has no bearing on optimal play.

3 Strategies

Catch-Up, for any finite set S, is a finite two-person zero-sum game of perfect information, so there exists a pair of optimal strategies such that (i) P_1 can guarantee a win, (ii) P_2 can guarantee a win, or (iii) the game is a draw. In order for a perfect-information game to be non-trivial, the optimal strategy should not be obvious to play.

In addition, different strategies should present a *heuristic tree* [1], such that there are some simple heuristics that new players can learn, and better performing but more complicated heuristics for more sophisticated players.

3.1 Optimal Play

We cannot yet prove whether Catch-Up($\{1, ..., N\}$) is a win, loss, or draw for P_1 for any N; however, for a given set, we can efficiently run a minimax algorithm with *alpha-beta pruning* and *transposition tables* [12] to solve the game value, assuming optimal play by both players. We have calculated the game values for Catch-Up($\{1, ..., N\}$) up to N=20. Results for optimal play are shown in Table 2 in the optimal play row, with -1 being a loss for P_1, 1 being a win for P_1, and 0 being a tie game.

As described in Section 2.6, Catch-Up($\{1, ..., N\}$) games of the form $N = 4n + 3$ or $N = 4n + 4$ permit draws. We have calculated that these games, up to at least $N = 20$, are draws for optimal play. We believe that this pattern holds for all n, though we have not been able to prove this and can only offer it as a conjecture. Using Monte-Carlo tree search [13], we have explored values of $N = 23, 24, 27$, and 28 and did not find any contradictions.

N	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
optimal play	0	0	1	1	0	0	-1	-1	0	0	1	-1	0	0	1	-1	0	0

Table 2. Optimal play values for Catch-Up($\{1, ..., N\}$) relative to P_1: 1 is a win, -1 is a loss, and 0 is a draw.

	P_2 *Random*	P_2 *MaxScore*	P_2 *MinScore*	P_2 *UseMostNums*
P_1 *Random*	50.01%	41.66%	60.87%	46.53%
P_1 *MaxScore*	63.04%	14.28%	29.78%	48.06%
P_1 *MinScore*	37.56%	79.74%	0.00%*	37.79%
P_1 *UseMostNums*	52.70%	49.73%	60.54%	50.87%

Table 3. Win percentages for P_1 when playing different heuristics against P_2 in Catch-Up($\{1, ..., 10\}$).

3.2 Human-Playable Heuristics

Although machines can efficiently search a game tree for optimal moves, humans do not think in the same way and, generally, do not find it enjoyable (or possible) to exhaustively explore every move when playing a game.

In order for a strategy to work for human players, we need effective heuristics that are accessible and can be easily used. And for a game to have lasting depth, simple heuristics must be generally less effective than more complex ones, so there is a benefit for continued study and improvement.

We analysed several simple human-usable heuristics for playing Catch-Up. For these heuristics, if multiple moves could be chosen, one of them is picked at random. We do not claim that these are the only heuristics for players, or that players should follow any of them. Instead, they provide a starting point for strategies that new players might try, which help us understand if the game can be enjoyed by beginners.

1. *Random*: Players choose any move at random.
2. *MaxScore*: Players maximise their scores on every turn, extending their leads by as much as possible.
3. *MinScore*: Players minimise their scores on every turn, keeping their scores as close as possible.
4. *UseMostNums*: Players use as many numbers as possible, reducing the numbers available for the opponent.

For Catch-Up($\{1, ..., 10\}$), which is an odd-sum game, optimal play by both players leads to a loss for P_1, but it is difficult for humans to play optimally. Instead, we can test the various simple heuristics and compare how they perform against each other.

For example, Table 3 shows the probabilities of P_1 winning when playing each of her heuristics against each of P_2's 100,000 times. The value in each cell indicates the percentage of games in which P_1 wins; a value of 1 means that P_1 always wins, whereas a value of 0 means that P_1 always loses. Values $>$ .5 in Table 3 are good for P_1, whereas values $<$ .5 are good for P_2. Players are assumed to use the same heuristic throughout the entire game, without switching or adapting within a game to what the other player is doing.

The P_1 *Random* vs P_2 *Random* cell shows that random play gives close to a 50% chance of winning, which indicates completely unskilled play will not favour one player over the other. Looking at the first column, we see the effect of P_1 using each heuristic against P_2 *Random*, and that P_1 *MaxScore* is the best of the four heuristics, improving P_1's win rate to approximately 63%, whereas P_1 *MinScore* is a bad heuristic, reducing the win rate to around 37.6%. Similarly, if we look at the first row, which shows the effect of P_2 using each heuristic against P_1 *Random*, we see that P_2 *MaxScore* is the best heuristic for P_2.

However, if both players adopt the *MaxScore* heuristic, this is bad for P_1, reducing P_1's win rate to around 14.3%. P_1, playing against a P_2 *MaxScore* heuristic, would do better to use the P_1 *MinScore* heuristic, which was previously a bad choice. But this can lead to P_2 in turn switching to the P_2 *MinScore* heuristic, in which P_2 now wins every game. Likewise, P_1 now does better by switching back to the P_1 *Random* heuristic.

Given these simple heuristics, we already see an interesting pattern, in which there is not one dominating heuristic. This is an indication that Catch-Up does not have a trivial or obvious solution for human players. We believe this *rock-paper-scissors* balance, in which different heuristics perform better in some cases but not others, but no one heuristic dominates, is an important characteristic of deep and interesting games.

These heuristics do not necessarily generalise to other sets S. Just because a heuristic does well in Catch-Up($\{1, ..., 10\}$) does not mean it does well in Catch-Up($\{1, ..., 9\}$), another odd-sum game. For example, P_1 *MinScore* vs P_2 *MaxScore* wins 79.7% for P_1 in the former game, but flips to only a 34.1% win rate for P_1 in the latter game. Clearly, these heuristics offer only a glimpse into optimal play of Catch-Up.

3.3 Climbing the Heuristics Tree

We can simulate a player adopting a new, better heuristic by combining the previous four heuristics. Instead of deciding between multiple moves randomly, we can apply a second-level heuristic to choose between multiple moves. For example, P_1 using *UseMostNums* + *MinScore* would first pick moves to use the most numbers, and if there is more than one remaining move to choose from, she chooses the move that sums to the smallest number. As before, any final remaining options are eliminated by selecting one at random.

If P_2 adopts this *UseMostNums* + *MinScore* combination heuristic, but P_1 stays with the original heuristics, P_2 now wins every game against two P_1 heuristics, and wins a slight majority of games otherwise, as shown in Table 4.

	P_2 *UseMostNums* + *MinScore*
P_1 *Random*	45.88%
P_1 *MaxScore*	0.00%
P_1 *MinScore*	0.00%
P_1 *UseMostNums*	45.55%

Table 4. P_2 win rate using combination heuristic.

Note that the purpose of this section is not to present the reader with the best heuristics, but to show that Catch-Up provides a compelling platform for developing effective heuristics for human play.

4 Properties of Catch-Up

In analysing the properties that follow, we do not always have analytical proofs for all S, or for all values N for $S = \{1, ..., N\}$, so we offer conjectures and computational analysis of games for relatively small N. Some of these metrics have been used to determine whether a game is well designed, which have been shown to be effective at generating new game designs [3].

4.1 Total Points Scored

Catch-Up ends only when all the numbers have been incorporated in either P_1's score s_1 or P_2's score s_2. For Catch-Up($\{1, ..., N\}$), the sum of the players' scores will be equal to the *triangular number* $T(N)$ [14, Seq. A000217] :

$$s_1 + s_2 = T(N) = \sum_{i=1}^{N} i = \frac{N(N+1)}{2} \qquad (1)$$

4.2 The Lead is Always $\leq \max(S)$

A player can never be winning by more than $M = \max(S)$, the largest number in S, which includes the final move. Thus, a designer can choose elements of S to force the game always to be within a range of M points.

This is relatively easy to prove. Let P_i be the acting player, and P_j be the opponent. A turn must end when a P_i ties or exceeds P_j's score, so right before choosing the last number that ends a turn, P_i must either be starting tied or be behind, so $s_i - s_j \leq 0$. The largest number that can possibly be chosen as the last selection on the turn is M. Thus, at the end of the current turn, the score difference can be no more than $M = \max(S)$.

4.3 Maximum Points Per Turn

We can analyse the maximum number of points that can be earned on a turn of Catch-Up($\{1, ..., N\}$). On P_1's first turn, she would choose N, the largest number available. On P_2's turn, he can first select numbers that sum to $N - 1$ (if P_2 were to exceed $N - 1$, then the turn would immediately end) plus the largest remaining number, $N - 1$. This can be done by selecting $\langle 1, N-2, N-1 \rangle$, which gives P_2 a maximum sum of $2N - 2$ on a single turn. Note that P_2 would also have achieved this if P_1 first chose $\langle N-1 \rangle$, and P_2 responded by choosing $\langle N-2, N \rangle$, also giving a total of $2N - 2$.

4.4 Game-Tree Size

The game-tree size gives the total number of unique play-throughs, iterating through all valid moves of the game. This is equivalent to counting the number of terminal nodes in the game tree. For simplicity, we consider each permutation of a player's removal choices in a single turn to be a distinct branch, although the order of removals within a turn does not matter during play.

Large game trees are more difficult for players to utilise in play, as they do not permit memorisation of the best moves; however, they also make it computationally harder for analysis by adversarial search. By increasing the size of the set S, the game tree rapidly increases in size.

The game-tree size is exactly $N!$, which is the number of ways the numbers in the set can be picked, and then assigning turns after determining the order, the numbers are picked to make it a valid game of Catch-Up. Table 5 enumerates all possible games of Catch-Up($\{1, ..., N\}$) for up to $N = 18$ and counts the number of terminal nodes, verifying the game-tree size is indeed $N!$.

4.5 State-Space Size

State-space size is the number of possible states of the game, reflecting the fact that many states can be reached from multiple moves [15]. This process converts the game tree into a directed acyclic graph, because a game state represented in the graph can have multiple parents.

N	game-tree size ($N!$)	state-space size	P_1 optimal play value	max branch factor	min depth	win %	tie %	loss %
3	6	11	0	3	2	16.67%	66.67%	16.67%
4	24	33	0	6	2	33.33%	33.33%	33.33%
5	120	90	1	16	3	50.00%	0.00%	50.00%
6	720	236	1	36	3	50.00%	0.00%	50.00%
7	5040	591	0	78	3	38.57%	22.86%	38.57%
8	40320	1453	0	150	4	38.93%	22.14%	38.93%
9	362880	3484	-1	272	4	50.00%	0.00%	50.00%
10	3628800	8174	-1	474	4	50.00%	0.00%	50.00%
11	39916800	18799	0	886	5	41.90%	16.19%	41.90%
12	479001600	42587	0	1470	5	42.47%	15.06%	42.47%
13	6227020800	95126	1	2448	5	50.00%	0.00%	50.00%
14	87178291200	210064	-1	3894	6	50.00%	0.00%	50.00%
15	1.30767E+12	459225	0	6158	6	43.86%	12.28%	43.86%
16	2.09228E+13	995349	0	10284	6	44.23%	11.53%	44.23%
17	3.55687E+14	2141652	1	16048	6	50.00%	0.00%	50.00%
18	6.40237E+15	4579850	-1	24762	7	50.00%	0.00%	50.00%

Table 5. Measures for Catch-Up($\{1, ..., N\}$) for values of $N = 3$ to $N = 18$.

In Catch-Up, the necessary states to track are: current player, current score and numbers remaining in the set.[4] We do not have an analytical bound for the state-space size, but empirical data generated for small N, shown in Table 5, demonstrates that it grows much more slowly than the game-tree size. For large N, the state-space size is much smaller because there are many ways to reach the same game state using different moves.

For example, for any Catch-Up($\{1, ..., N\}$) for $N \geq 3$, the following game traces all reach an identical game state with tied score $3-3$: $(\langle 3\rangle, \langle 1,2\rangle)$; $(\langle 3\rangle, \langle 2,1\rangle)$; $(\langle 2\rangle, \langle 3\rangle, \langle 1\rangle)$; $(\langle 1\rangle, \langle 3\rangle, \langle 2\rangle)$. Thus, huge benefits occur from caching results in a transposition table [12] when exploring the game graph for optimal moves.

4.6 Game-Tree Depth

The depth of a game tree for Catch-Up(S) can be no deeper than $|S|$ turns. Thereby, the designer or players can control the length of the game by choosing the size of S.

This maximal depth occurs when each player selects the smallest number in the set, with each ending a turn with only one number removed. This gives a total of N turns. Games can certainly end sooner, because on some turns a player may select more than one number, decreasing the number of turns for that path in the tree.

The minimum length of the game is also determined by the size of the initial set. We do not have an analytical lower bound for Catch-Up($\{1, ..., N\}$), but we present calculated minimum-depth values in Table 5.

4.7 Maximum Selections per Turn

For Catch-Up($\{1, ..., N\}$), we can calculate K, the maximum number of numbers that can be selected on a turn. This can help a designer understand how long a turn will take for players to evaluate. We show in Appendix 5.2 that K is $O(\sqrt{N})$ and has an exact analytical value of:

$$K = \lfloor \sqrt{2N - 7/4} + 1/2 \rfloor$$

4.8 Branching Factor

The maximum branching factor, which we call B_{max}, tells us how many possible moves there are on a turn in the worst case. The higher the branching factor, the more complicated a game can be for a player to explore. The maximum branching factor for Catch-Up($\{1, ..., N\}$) is:

$$B_{max} = O\left(N^{\sqrt{2N}+1}\right)$$

A derivation of this upper bound is provided in Appendix 5.3.

In Table 5, we show the empirical maximum branching factor, which is the maximum of the number of first moves by P_1 and the number of replies (to first moves) by P_2. This table clearly shows that the maximum branching factor is exceedingly high for a game, making it difficult to explore the entire early game tree for large N.

As Catch-Up proceeds, there are fewer numbers in S to choose from, so the branching factor B_t for each turn t will decrease until the final move, which forces the last player to select all remaining numbers. The average branching factor B_{avg} will be less at each layer of the game tree.

[4] If this information is known, it does not matter how the removed numbers were chosen to get to this state.

Turn	1	2	3	4	5	6	7	8	9	10	11	12
B_{avg}	12	366.67	35.42	15.52	7.80	4.32	2.72	1.94	1.52	1.27	1.11	1.00
B_{max}	12	1470	738	738	560	560	258	108	24	6	2	1

Table 6. Average (B_{avg}) and maximum (B_{max}) branching factors for Catch-Up($\{1, ..., 12\}$).

We do not have an analytical bound for B_{avg}, but can calculate it empirically for small N, allowing us to generate the entire tree.

In Table 6, we give the average and maximum branching factors per level of the game tree for, as an example, Catch-Up($\{1, ..., 12\}$). Level l of the tree represents the possible game states and moves available on turn l. To calculate the average branching factor, we expand the entire game tree and then calculate how many moves there are available on each level of the tree divided by the number of unique states on that level. The average and maximum branching factors peak on Turn 2 (P_2's first turn) and then rapidly decrease as the game progresses to the end.

4.9 Win/Loss/Draw Ratios

It is useful to understand if a game is balanced by looking at a game's win/loss/draw ratios. For small N, we can analyse the entire game tree to calculate the percentage of wins, losses, and draws. The Win %, Tie %, and Loss % columns in Table 5 show the results of exploring all possible games of Catch-Up($\{1, ..., N\}$) from $N = 3$ to $N = 18$. As explained in Section 2.6, tie games are impossible in games where $T(N)$ is odd, and these tie percentages are indicated as 0.00%. As N increases, the chance of a random game ending in a draw decreases, which suggests that the games are not too 'drawish'. We also note the games are balanced between P_1 and P_2, suggesting that there is no inherent advantage in going first or second if not playing optimally.

4.10 Solutions that Lead to a Draw

For Catch-Up($\{1, ..., N\}$), it is also possible to enumerate the moves which will lead to a draw by finding the assignment of positive and negative numbers using the following equation:

$$1 \pm 2 \pm ... \pm N = 0 \qquad (2)$$

Assigning positive numbers to P_1 and negative numbers to P_2 gives us all possible solutions that lead to a draw.

The number of unique assignments of plus and minus for large N can be calculated using a generating function [14, Seqs. A063865, A058377], which was first discovered by Euler and has been shown to have an asymptotic upper bound of $\sqrt{6/\pi} * N^{-3/2} * 2^N$.

One way to solve this is to find *strict partitions* of $T(N)/2$. Strict partitions are sets of nonrepeating integers that add up to a given sum; for example, a strict partition of 10 is $\{2, 3, 5\}$ or $\{1, 3, 6\}$ but not $\{1, 1, 3, 5\}$ or $\{2, 2, 2, 2, 2\}$. We can use a strict partition to give the unique integers that can sum to half the total score for the game, which is the condition for a draw. One can generate partitions [16] and then remove the ones with repeated integers, or generate them directly using generating functions [14, Seq. A000009].

Every solution to Equation 2 can be reached by having each player choose the smallest numbers in their assigned partition until their turns end, although the resulting draw by this method is likely not to be the result of optimal play.

5 Conclusion

One of the most interesting properties of Catch-Up is the complexity of the game tree, given its minimal game rules. This makes it challenging for players to calculate optimal moves by backward induction and adversarial search, necessitating the use of heuristics to play the game. Catch-Up rules not only can encourage drama and tension in games, but they also have interesting mathematically emergent properties. We hope this analysis provides game designers with alternative ways of thinking about using catch-up mechanisms in their own games.

In two-player or multiplayer games, the players' current scores often provide a clue as to how well they are doing in comparison to the other players. One interesting aspect of Catch-Up is that until the last few moves, the scores do not provide this information because the lead switches on every turn (except for ties). Thus, players need to generate other methods of evaluating the state of the game so that they can tell if they are ahead or behind, but these are not obvious in a game like Catch-Up. Players accustomed to treating current scores as an indication of who is winning may find this to be an interesting feature, or an unpleasant surprise.

We believe that our work on Catch-Up offers lessons that might help guide designers when constructing their own games:

- The structure and form of the pieces can greatly change how a game is perceived. The accessibility of the physical version facilitates play because it does not require the players to keep track of their scores.

- Catch-up mechanisms are intended to keep players feeling that the game is close. In combinatorial games, however, it can be disguising the actual state of the game and the likelihood of each player to win, lose, or draw.

- The starting conditions for a game – the set S in Catch-Up – can have a huge impact on the solution space and play dynamics.

- Simple heuristics are easy to implement in software and can help determine if new players can successfully compete.

- Proving if a game has good characteristics is often significantly more difficult than simulating them; yet much can still be learned from simulating game play.

We conclude by posing several open questions for future study:

- What sets of numbers S are most enjoyable for players?

- How do repeated or non-consecutive numbers in S change the game and the properties we have analysed?

- Is it possible to prove our conjecture that optimal play always leads to a draw in even-sum games of Catch-Up($\{1, ..., N\}$)?

- What is the analytical bound for state-space size? Is there a better upper bound for the branching factor?

- Is multiplayer Catch-Up, where the player with the lowest score goes next, a playable game with interesting properties?

- Which heuristics are most effective across different sets S in Catch-Up?

- What are the most interesting ways to break ties? For example, one could break a tie by comparing excess sums, calculated by summing the leads that players achieved on their turns. By not allowing ties, one can do a Nim-like analysis of Catch-Up by treating it as an impartial game, using the Sprague-Grundy theorem [17].

- What happens if one of the players starts with a non-zero score? One could start P_2 with a positive score, so P_1 moves first but starts from behind. This changes the analysis of odd-sum games of Catch-Up($\{1, ..., N\}$) such that they may end in draws.

Acknowledgements

Thanks to Dan Gopstein for comments, suggesting the physical version for Catch-Up and proposing a multiplayer variant; to Kaho Abe for help in 3-D printing a version of the game; to Frank Lantz for various suggestions. We would like especially to thank the anonymous reviewers and Editor-in-Chief for their detailed, thoughtful, and helpful corrections, comments, and recommendations, as well as for pointing out several existing games (including Zhang-Qi) that use catch-up mechanisms which we were not aware of. Also thanks to Dan Isaksen for discussions on integer partitioning theory and for ideas on strategies to find an analytical proof for the game value of Catch-Up($\{1, ..., N\}$). Part of this research has been done during Mehmet Ismail's visit to NYU; he would like to thank the Department of Politics for its hospitality.

References

[1] Elias, G. S., Garfield, R., Gutschera, K. R. and Whitley, P., *Characteristics of Games*, Massachusetts, MIT Press, 2012.

[2] Thompson, J. M., 'Defining The Abstract', *Game & Puzzle Design*, vol. 1, no. 1, 2015, pp. 83–86.

[3] Browne, C. and Maire, F., 'Evolutionary Game Design', *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 2, no. 1, 2010, pp. 1–16.

[4] Fehr, E. and Schmidt, K. M., 'A Theory of Fairness, Competition, and Cooperation', *Quarterly Journal of Economics*, vol. 114, no. 3, 1999, pp. 817–868.

[5] Nealen, A., Saltsman, A. and Boxerman, E., 'Towards Minimalist Game Design', in *Proceedings of the 6th International Conference on Foundations of Digital Games (FDG 2011)*, New York, ACM Press, 2011, pp. 38–45.

[6] Gardner, M., *The Colossal Book of Mathematics: Classic Puzzles, Paradoxes, and Problems: Number Theory, Algebra, Geometry, Probability, Topology, Game Theory, Infinity, and Other Topics of Recreational Mathematics*, New York, WW Norton & Company, 2001.

[7] Berlekamp, E. R., Conway, J. H. and Guy, R. K., *Winning Ways for Your Mathematical Plays*, vol. 1, Natick, AK Peters, 2001.

[8] Parlett, D., *Hare & Tortoise*, London, Intellect Games UK, 1974.

[9] Bauza, A., *Tokaido*, Correggio, Asterion Press, 2012.

[10] Miyamoto, S., *Super Mario Kart*, New York, Nintendo, 1992.

[11] Kahneman, D., 'Maps of Bounded Rationality: A Perspective on Intuitive Judgement and Choice', *Nobel Prize Lecture*, 2002.

[12] Russell, S. and Norvig, P., *Artificial Intelligence: A Modern Approach*, second edition, New York, Prentice Hall, 2009, pp. 170–171.

[13] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S. and Colton, S., 'A Survey of Monte Carlo Tree Search Methods', *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, 2012, pp. 1–43.

[14] 'A000217: Triangular Numbers', *The On-Line Encyclopedia of Integer Sequences*, 2010. https://oeis.org/A000217

[15] Allis, L. V., *Searching for Solutions in Games and Artificial Intelligence*, PhD thesis, University of Limburg, Maastricht, 1994, pp. 158–169.

[16] Kelleher, J. and O'Sullivan, B. 'Generating All Partitions: A Comparison Of Two Encodings', tech. report, 2014. http://arxiv.org/abs/0909.2331v2

[17] Ferguson, T. S., *Game Theory*, second edition, part I, Los Angeles, UCLA, 2014, Part I, p. 22.

Aaron Isaksen is a PhD Candidate in Computer Science at the NYU Tandon School of Engineering, researching computer-aided game design, and a professional independent video game developer. **Address:** NYU Tandon School of Engineering, 6 MetroTech Center, Brooklyn, NY 11201, USA. **Email:** aisaksen@nyu.edu

Mehmet Ismail is a PhD Candidate in Economics at Maastricht University, whose research interests include game theory, decision theory and behavioral economics. **Address:** Department of Economics, Maastricht University, PO Box 616, 6200 MD Maastricht, The Netherlands. **Email:** mehmet.s.ismail@gmail.com

Steven J. Brams is Professor of Politics at New York University, whose research interests are game theory, social choice theory and fair division, with application to international relations, voting and elections, literature, history, philosophy and religion. **Address:** Department of Politics, 19 West 4th St, New York University, NY 10012, USA. **Email:** steven.brams@nyu.edu

Andy Nealen is Assistant Professor of Computer Science at the NYU Tandon School of Engineering, co-director of the Game Innovation Lab, and co-creator of the video game Osmos. His research interests are in game design and engineering, computer graphics and perceptual science. **Address:** NYU Tandon School of Engineering, 6 MetroTech Center, Brooklyn, NY 11201, USA. **Email:** nealen@nyu.edu

Appendix

5.1 The Existence of Draws

For Catch-Up($\{1, ..., N\}$), we prove which games will permit draws and which enforce a winner, based on the value of N.

To begin, the final score of both players in Catch-Up will always add to the sum of all the numbers in S, because the game will only end once all numbers are assigned to either P_1 or P_2. From Equation 1, the total score $T(N)$ for a game with a set $S = \{1, ..., N\}$ is $N(N+1)/2$.

The key factor here is to determine if the sum $T(N)$ is even or odd. If $T(N)$ is odd, such that $T(N) \bmod 2 = 1$, then there is no way to partition S into two subsets S_1 and S_2 such that the final scores are equal. If $T(N)$ is even, such that $T(N) \bmod 2 = 0$, then there is a way to assign the numbers such that the players have equal scores at the end.

If we write $N = 4n + k$, where $n \geq 0$ and $k \in \{1, 2, 3, 4\}$, we can determine, for all N, which games will have even and odd sums:

$$\begin{aligned}&T(4n+k) \bmod 2\\ &= (4n+k)(4n+k+1)/2 \bmod 2\\ &= 8n^2 + 4nk + 2n + k^2/2 + k/2 \bmod 2\\ &= \tfrac{k^2+k}{2} \bmod 2\end{aligned}$$

Thus, parity is independent of n, and we can show if it is odd or even for each $k \in \{1, 2, 3, 4\}$:

$$\begin{aligned}k = 1 &: \tfrac{1^2+1}{2} \bmod 2 = 1\ (odd)\\ k = 2 &: \tfrac{2^2+2}{2} \bmod 2 = 1\ (odd)\\ k = 3 &: \tfrac{3^2+3}{2} \bmod 2 = 0\ (even)\\ k = 4 &: \tfrac{4^2+4}{2} \bmod 2 = 0\ (even)\end{aligned}$$

Therefore, games of the form $N = 4n + 1$ and $4n + 2$ will always have a winner by at least one point, and games of the form $N = 4n + 3$ or $4n + 4$ can, but are not required to, end in a draw.

5.2 Maximum Selections per Turn

We prove the claims of Section 4.7 to calculate K, the maximum number of numbers that can be selected on a turn. On the first turn, no matter what set S is, P_1 can only select one number. P_2 can then choose from $N - 1$ numbers.

From Section 4.3, we know that the greatest sum of numbers that can be earned on a turn is $2N - 2$; before the last number is selected on the turn, the sum of points earned can be no more than $N - 1$. We want to find the maximum number of selections that sum to $N - 1$, and then add 1 for the final selection that ends the turn.

The maximum number of selections occurs if the player selects $\{1, 2, ..., k\}$ such that the sum $1 + 2 + ... + k$ is as large as possible while still $\leq N - 1$:

$$1 + 2 + ... + k = \frac{k(k+1)}{2} \leq N - 1$$

This is quadratic in k, so we can use the quadratic formula with $a = 1, b = 1, c = -2N + 2$ to find the positive k that maximises the sum. Adding 1 for the final number that takes the sum to $\geq N$ to end the turn, we have the maximum number of selections on a turn $K = k + 1$ as:

$$K = \lfloor \sqrt{2N - 7/4} + 1/2 \rfloor$$

which is $O(\sqrt{N})$ since $K < \sqrt{2N} + 1/2$.

5.3 Branching Factor (Derivation)

We find an upper bound for the maximum branching factor B_{max} for Catch-Up($\{1, ..., N\}$) as follows:

On the first turn, P_1 can only select one number, so the branching factor for turn 1 is N. In general, when playing with any set S, the first-turn branching factor is $|S|$.

For the remaining turns, we can calculate an upper bound B_{max} for the maximum branching factor for Catch-Up($\{1, ..., N\}$) based on the results of Section 4.7. A player can select at least one number and at most K numbers on a turn, and those numbers can be permuted except for the final one selected, so we have an upper bound for the branching factor as:

$$B_{max} < \sum_{i=1}^{K} \binom{N-1}{i} (i-1)! < \sum_{i=1}^{K} \binom{N}{i} i!$$

Because $\binom{N}{i} i! = \frac{N! i!}{(N-i)! i!} = \frac{N!}{(N-i)!} < N^i$ and generally $K < N/2$ since $K = O(\sqrt{N})$, we have $\binom{N}{i} \leq \binom{N}{K}$ for $i \leq K$, and therefore:

$$B_{max} < \sum_{i=1}^{K} N^i < \sum_{i=1}^{K} N^K < KN^K$$

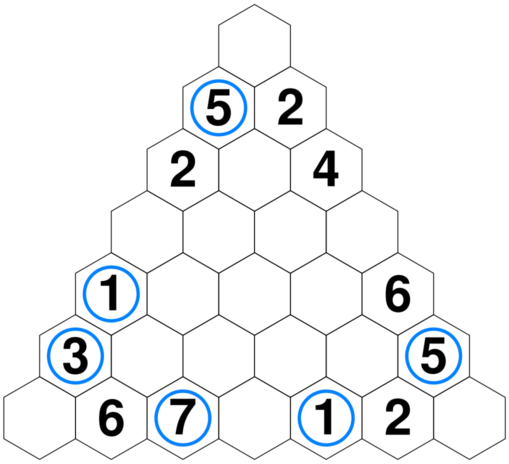
Thus, we have the final upper bound for the maximum branching factor:

$$B_{max} < (\sqrt{2N} + 1/2) N^{(\sqrt{2N} + 1/2)}$$

$$B_{max} = O\left(N^{\sqrt{2N}+1}\right)$$

Try Challenges #7 and #8

Fill the grid with numbers 1 to 7, such that no number is repeated along any orthogonal line, and no connected group of odd numbers touches all three sides. See p. 21 for details.


5
2
2
4
1
6
3
5
6
7
1
2