POLISH—OR—LET'S PLAY THE CLEANING GAME

PRZEMYSŁAW GORDINOWICZ, RICHARD J. NOWAKOWSKI, AND PAWEŁ PRAŁAT

ABSTRACT. POLISH is a game based on the 'Cleaning with Brushes' model. It is a combinatorial game in the sense of Conway but can be seen as a graph searching or chip-firing problem as well. We consider only the impartial version and give a characterization of graphs with maximum degree two that are first player wins. We also show that the second player can win on the complete graph K_n provided $n \geq 3$ and the complete bipartite graph $K_{1,n}$ and $K_{2,n}$ provided $n \neq 1, 3$. We give the nim-values for all positions on paths, stars, and also the nim-values for complete bipartite graphs where every vertex needs at most 3 more brushes to fire.

1. INTRODUCTION

We introduce POLISH[1], an *impartial* game played on a finite simple graph. This game is a modification of the one-player 'game' Brush Cleaning, introduced in [15] also see [2, 8, 9, 16, 17, 18, 19]. The Brush Cleaning Model is played on a finite graph where some vertices have *brushes*. If the number of brushes on vertex x equals or exceeds the degree of x, we say x is *primed*. If this is the case, then x *fires*, that is, one brush is sent along each incident edge after which x and all its incident edges are *cleaned* and deleted. If one or more vertices is primed, choose one and fire it and continue firing vertices until no vertex is primed. It can be shown that the order of firing does not affect the final set of vertices that are not primed. The object is to determine the minimum number of brushes and their placement, needed to clean the graph. It is similar to the chip-firing model [5] and also the decontamination metaphor for graph searching [7] but it was motivated by several real life situations [12, 14]. Moreover, it was proved to be equivalent to the balanced vertex-ordering problem that plays an important role in graph drawing theory [4, 8, 13].

For POLISH, there are two players who play alternately and on their turn, place a brush on a vertex. The brushes are indistinguishable. If, after a placement, a vertex is *primed* then it is fired. If, after firing, one or more other vertices become primed, then, as before, one of them is chosen to fire and this continues until no vertex is primed. In particular, since every isolated vertex is primed, this algorithm deletes from the graph any vertices that become isolated. When the process stops, the next player

1991 *Mathematics Subject Classification.* 05C57, 91A43, 91A46.

Key words and phrases. cleaning game, chip firing, combinatorial game, impartial game, nim, brush number.

The second author gratefully acknowledge support from NSERC. The third author is supported by MPrime and Ryerson University.

[1]There is a long tradition of puns in the combinatorial games literature. We will leave the reader to determine how to say the name.

can introduce one more brush to the game. The player who cannot place a brush (because there is no vertex left) loses. Phrased positively, the player who cleans the last edge/vertex wins. FRENCH POLISH, the partizan version in which the brushes are different, we leave for another paper [10].

An important fact is that the order in which the primed vertices are fired is irrelevant. The following result was obtained in [15] (Theorem 2.1). In the graph cleaning literature, e.g. [15], at the start, all edges and vertices are called *dirty.* As the process progresses, vertices that are fired, and all incident edges, are called *clean.* Here, since we are not re-cleaning the graph, we delete clean edges and vertices.

Theorem 1 ([15])**.** *Given a graph $G = (V, E)$ and an initial configuration of brushes, the cleaning algorithm returns a unique final set of dirty vertices.*

In fact, we need slightly stronger statement. We need to make sure that the order of vertices that we fire does not affect a final configuration of brushes.

Theorem 2. *Given a graph $G = (V, E)$ and an initial configuration of brushes, the cleaning algorithm returns a unique configuration of brushes on the dirty vertices.*

Proof. Since the final configuration of dirty vertices is unique by Theorem 1, the number of brushes on any remaining dirty vertex is the sum of the number of brushes placed on this vertex plus the number of adjacent vertices that have been fired. □

Given a graph G, the main question to be answered is: *Can the first player force a win?* Before reading further, the reader might like to solve the following problem. Let G be the disjoint union of $K_{1,3}$, C_4, K_2 and $K_{2,3}$ and no brushes have yet been placed. Find the winning move on G!

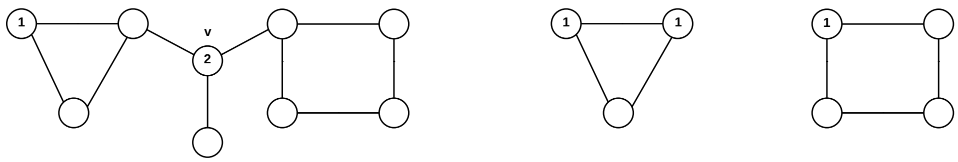
David Singmaster [20] showed that almost all games are first-player wins but this is a probabilistic result. When dealing with graphs where all the degrees have the same parity, it is not too surprising that the answer to 'who wins?' often depends on that parity. It is quite surprising, then, that in POLISH the second player wins on K_n for all $n \geq 3$ (Theorem 9) and $K_{2,n}$ for $n \notin \{1, 3\}$ (Theorem 13). Theorems 7 and 8 answer the question for a collection of cycles, paths, and stars, respectively. The 'collection' part turns the game into a disjunctive sum of individual games (each on a connected component) and this requires us to invoke Grundy values. We introduce these and other necessary combinatorial game theory notions in the next section. We still do not know everything about complete bipartite graphs but partial results and conjectures can be found in Sections 4, 5, 7 and 8. In the cases of playing on a star, or playing LIGHT POLISH on a complete bipartite graph, the situation can be described as playing with a token on a grid (lattice or $(\mathbb{Z}^{\geq 0})^n$), where each move takes the token toward the origin. The tables of values suggest that the recently developed lattice point methods [11] would be useful. Unfortunately, these games satisfy all the necessary conditions but one, our games includes an infinity of moves to terminal positions. Consequently, the proofs are *ad hoc* and use case-by-case analysis.

The paper is structured as follows. A brief introduction to combinatorial games is presented in Section 2. Section 4 gives the analysis for all graphs with maximum degree

2 and any number of brushes. Already, analyzing graphs with maximum degree 3 seems difficult. Stars and subdivided stars are investigated in Section 5. Complete graphs and complete bipartite graphs are analyzed in Subsections 6 and 7, respectively. We finish the paper with LIGHT POLISH an interesting variant of the game that is inspired by the game of 'Traffic Lights'—see Section 8.

2. Necessary background

In an impartial game (if we are only interested in 'Who wins?'), given a position G, we use the terms *next* and *first* player to mean the player who moves next (or first) in G. The other player will be called the *previous* or *second* player.



(a) before v is fired (b) after v is fired

FIGURE 1. Add a Brush to v and then fire v.

If a graph is sparse, then often during the play, the resulting graph will be disconnected. In Figure 1(a) for example, there are two brushes at vertex v. If another brush is added to v, then it is primed and Figure 1(b) is the result after firing. From this point on, at every round of the game, players have to choose one component to play on. In combinatorial game theory terms, it becomes a disjunctive sum (see below for a formal definition). When disjunctive sums occur, the description of the set of positions that the next player can win can be unwieldy. The Sprague-Grundy theory for impartial games (see [1] Chapter 7 or either of [3, 6]) is ideal for clarifying the description in these cases. Let S be a finite set of non-negative integers. Then $\operatorname{mex}(S)$ is defined as the least non-negative integer not contained in S. For example, $\operatorname{mex}\{0,1,3\} = 2$ and $\operatorname{mex}\{\,\} = 0 = \operatorname{mex}\{1,2,3,7\}$. In an *impartial game*, both players have the same moves in every position. An *option* of a position G is a position that can be reached in one move from G. The *Grundy value* of a position G in an impartial game is given by

$$\mathcal{G}(G) = \operatorname{mex}\{\mathcal{G}(G') : \text{ where } G' \text{ is an option of } G\}.$$

Note that if G is a *terminal* position, that is, G has no options, then $\mathcal{G}(G) = 0$. The *disjunctive sum* of positions G and H, written $G + H$, is the position in which a player may move in G or in H but not both at the same time. The XOR of non-negative numbers a and b, written $a \oplus b$, is defined as writing the numbers in a binary notation and then adding without carrying. XOR is also associative and commutative. For example, $3 \oplus 6 = 11_2 \oplus 110_2 = 101_2 = 5$ and $1 \oplus (3 \oplus 6) = (3 \oplus 6) \oplus 1 = 3 \oplus (6 \oplus 1) = 4$.

Almost all the proofs are 'By induction'. The induction is based on the length of the longest path in the game tree of that position. Every move reduces this length. Almost always the base case is the empty set and almost every statement about the empty set is true. (This makes top-down induction a useful tool. See [1], Appendix A, for example.)

The following two lemmas are necessary for any analysis of impartial games where the position include disjunctive sums. The second lemma says that to understand a disjunctive sum, we need only analyze the individual components, find their Grundy values and then XOR these values. If the result is 0 then the first lemma says that the second player can force a win in the sum; otherwise the first player has a winning strategy.

Lemma 3 ([1, 3, 6])**.** *A position G of an impartial game is a second player win if and only if $\mathcal{G}(G) = 0$.*

Lemma 4 ([1, 3, 6])**.** *Let G and H be positions in an impartial game. Then,*

$$\mathcal{G}(G + H) = \mathcal{G}(G) \oplus \mathcal{G}(H).$$

One last word on notation. The game of NIM played with one heap of counters has the simple rule that, on their turn, a player can take away as many counters as they wish. The player faced with the empty heap loses. Given an impartial game G, let $g = \mathcal{G}(G)$. The disjunctive sum of a NIM heap of size g and G is a second player win, moreover, no other heap sizes have this property. That is, playing on G is equivalent to playing on a NIM heap of size g. Because of this equivalence, in more recent combinatorial game papers, the term Grundy number or Grundy value has been replaced by the term *nim-value* and we shall use the latter terminology.

3. A general result.

Parity does play a part in the analysis of POLISH as the next theorem shows. However, parity does not give a complete characterization of the state of play.

Let G be a graph and x a vertex of G. Let $\Delta(x)$ denote the *discrepancy* of x, that is the degree of x minus the number of brushes at x. In other words, this is the number of brushes needed for x to be primed.

Theorem 5. *Let G be a graph with a number of brushes. If the discrepancy of every vertex is even then $\mathcal{G}(G) = 0$.*

Proof. Since $\mathcal{G}(G) = 0$ is equivalent to the second player winning, we are going to show that a copy-cat strategy for the second player is a winning one.

If G is empty then all discrepancies are even and G is also a second player win. If G is not empty then a move by the first player, say on vertex x, changes the parity, to odd, of the discrepancy of x. Since $\Delta(x) \geq 1$, x is not primed. The second player places another brush at x. If now $\Delta(x) \geq 2$ then all discrepancies are even and the second player wins by induction. If $\Delta(x) = 0$ then it is primed and fires. Consider an adjacent vertex, y, of x. It gets an extra brush and the incident edge, that brush traversed, is deleted. Thus the discrepancy of y is has been decreased by 2, the parity has not changed. Even if more vertices become primed, the act of firing does not change the discrepancies of any vertices. Thus the second player has created a graph with all even discrepancies. □

This is not a characterization of the second player win positions. Consider the graph with two disjoint edges. All discrepancies are odd but the graph is a second player win.

'An even number of odd discrepancies' is also not a characterization. The graph $K_{1,n}$ where the central vertex has discrepancy 3, has $n+1$ vertices with odd discrepancy and, as we shall see in Section 5, it is a second player win regardless of the parity of n.

4. Graphs with Maximum degree 2

The only graphs to be considered in this section are paths and cycles. We may be looking at a position that results from playing in a larger graph, consequently, each connected subgraph may contain brushes.

Call a path *even* if the number of degree 2 vertices without a brush is even; otherwise it is called *odd*. A cycle with no brush is called *neutral*; a cycle with at least one brush it is called *even* if there is an even number of degree 2 vertices without a brush and it is called *odd* otherwise. From Theorem 5 we know that an even cycle has nim-value 0 and can be ignored but we include such subgraphs in the next lemma for completeness.

Lemma 6. *Let G be a non-empty connected graph with maximum degree 2. If G is a neutral cycle then $\mathcal{G}(G) = 0$; if G is an even path or even cycle then $\mathcal{G}(G) = 1$; else if G is odd then $\mathcal{G}(G) = 2$.*

Proof. By induction. Recall that we have to show that, from G, there is no move that results in a position with the claimed nim-value and that if m is a non-negative integer less than the claimed nim-value, there is a move that results in a position with nim-value m.

If G is empty then it has no options and so $\mathcal{G}(G) = \text{mex}\{\ \} = 0$. If G is a neutral cycle then $\mathcal{G}(G) = 0$ by Theorem 5.

In all other cases, note that there is always a move on a leaf vertex or on a vertex with a brush. This move results in cleaning the component and is a move to 0. All other moves are on nodes with no brushes and swap the parity of the component. Hence, if G is even, $\mathcal{G} = \text{mex}\{0, 2\} = 1$, while if G is odd then $\mathcal{G}(G) = \text{mex}\{0, 1\} = 2$. □

Theorem 7. *Let G be a collection of cycles and paths each possibly containing brushes. The previous player can force a win if the number of even components and the number of odd components are both even, otherwise the next player can force a win.*

Proof. The neutral cycles contribute 0 to the XOR of the disjunctive sum and so can be ignored. Let C_e and C_o be the set of even and odd connected components respectively. By Lemma 4, if $A, B \in C_i$, $i \in \{o, e\}$ then

$$\mathcal{G}(A + B) = \mathcal{G}(A) \oplus \mathcal{G}(B) = g \oplus g = 0$$

so only the parity of the sets is important.

If $|C_e|$ and $|C_o|$ are both even then the nim-value of the disjunctive sum is 0. If $|C_e|$ is even and $|C_o|$ is odd then the nim-value of the disjunctive sum is 2 and the winning move is to completely clean one of the odd components. If $|C_e|$ is odd and $|C_o|$ is even then the nim-value of the disjunctive sum is 1 and the winning move is to completely clean one of the even components. If $|C_e|$ and$|C_o|$ are odd then the nim-value of the disjunctive sum is $1 \oplus 2 = 3$ and the winning move is to convert an odd component to an even component by adding a brush to a vertex with no brush since the nim-value would now be $1 \oplus 1 = 0$. □

5. Cleaning stars

A *star* is $K_{1,n}$ for some n. Given a star, we will denote the position by (c, l) where c is the discrepancy at the central vertex and l is the number of leaves remaining. From (c, l), there are two moves: one move is to add a brush at the central vertex giving $(c-1, l)$, or to $(0,0)$ if $c = 1$; or add a brush to a leaf which, since it is now primed, will fire giving $(c-2, l-1)$ or $(0,0)$ if $c \leq 2$. At the beginning of the game on $K_{1,n}$ the position will be (n, n) and thereafter $c \leq l$.

Theorem 8. *The position* (c, l) *has nim-value* $c \pmod 3$*.*

Proof. We prove the result by induction. If $c = 0$ then the game is over since the central vertex will have fired and cleaned the graph. If $c = 1$ then both moves clean the graph so $\mathcal{G}((1, l)) = \text{mex}\{0\} = 1$. If $c > 1$ then the two moves give positions with $(c-1, l)$ and $(c-2, l-1)$. Hence $\mathcal{G}((c, l)) = \text{mex}\{c - 1 \pmod 3, c - 2 \pmod 3\} = c \pmod 3$. $\square$

What about Subdivided Stars?

Let $P_1, P_2, \ldots, P_n$, be a collection of disjoint paths and let c be a vertex disjoint from the paths. Make one leaf of each path adjacent to c. We call the resulting graph a *spider*. POLISH on a spider is equivalent to a game played with heaps of counters. A path of length $k+1$ is equivalent to a heap of size k, the central vertex is equivalent to its discrepancy, at the outset when there are no brushes, this would be a heap of size n. A move is to take one from 'arm' heaps (add a brush to an interior vertex) or remove the whole 'arm' heap (add a brush to the leaf). In the latter case, 2 are removed from the 'central' heap. Another move is to remove 1 from the 'central' heap (add a brush to c). The game is over when the 'central' heap becomes 0 (or less). Without the 'central' heap this is an easy game to analyze, with the central heap it appears to be very hard.

Even if each P_i is a path of two vertices matters become complicated. This game has an interesting mix of behaviours with some periodicity happening modulo 3, 4 and 12. We will denote the state of the game by a triple (c, m, l), where c is size of the 'central' heap (the discrepancy of the central vertex), m is the number of 'arm' heaps of size 1 (arm contains 1 brush), and l number of 'arm' heaps of size 2 (the number of arms with no brushes). A legal move then will be one of the following:

$$(c, m, l) \to \{(c-1, m, l), (c-2, m-1, l), (c-2, m, l-1), (c, m+1, l-1)\},$$

provided that the second and the third coordinate remain non-negative. Let $c = 4k + i$, $0 \leq i \leq 3$. The values fall into three regions: (1) $m \leq k$; (2) $l \leq k$; and (3) $m > k$, $l > k$. Let $\delta_{k,l} = 0$ if k and l are of the same parity otherwise $\delta_{k,l} = 1$. In region (3), from numerical calculations, it seems clear that:

- $(c, m, l) = (4k, m, l)$ has nim-value $0 + 3\delta_{k,l}$;
- $(c, m, l) = (4k + 1, m, l)$ has nim-value $1 + \delta_{k,l}$;
- $(c, m, l) = (4k + 2, m, l)$ has nim-value $2 - \delta_{k,l}$;
- $(c, m, l) = (4k + 3, m, l)$ has nim-value $0 + 3\delta_{k,l}$.

Unfortunately, the other regions require 12 subcases each. Moreover, the proofs for the nim-values also need to take into account the boundaries between region (3) and the other two. Restricting ourselves to just the question 'Who wins?' still needs all these sub-cases.

Question: For spiders, find a succinct way of describing the nim-values.

It is possible that an extension of the lattice point methods of [11] would completely solve spiders in a straightforward way.

6. Cleaning complete graphs

In this section we investigate POLISH played on a complete graph K_n for any integer $n \geq 2$. It is clear that the game played on K_2 is a first player win game. However, this is the only non-trivial complete graph that has this property.

Theorem 9. POLISH *played on K_n is a second player win for any $n \geq 3$.*

We prove this theorem independently depending on the parity of n (see Corollary 10 and Lemma 11 below). In the odd case, all the discrepancies are even, so this is a simple corollary of Theorem 5.

Corollary 10. POLISH *played on K_{2n+1} is a second player win for any $n \geq 1$.*

Now, it is time to move to the more complicated even case. Let us start by introducing the following notation. Let G be any complete graph on n vertices. Suppose that at some stage of the game played on G there still remain $k \leq n$ "dirty" vertices $v_1, v_2, \ldots, v_k$. This stage of the game can be described by the vector $(x_1, x_2, \ldots, x_k)$, where x_i denotes the number of brushes at v_i. Moreover, since vertices are indistinguishable, we assume, without loss of generality, that both the following properties hold: (i) all vector coordinates are sorted; that is, $x_i \geq x_{i+1}$ for $1 \leq i < k$; (ii) if there are vertices with the same number of brushes and we would like to play on them, we always add a brush to the first such coordinate; that is, every move increases by 1 exactly one x_i such that $i = 1$ or $x_{i-1} > x_i$ for some $2 \leq i \leq k$.

Lemma 11. POLISH *on K_{2n} is a second player win for any $n \geq 2$.*

Proof. At any point of the game played on K_{2n} that can be described by a vector $X = (x_1, x_2, \ldots, x_k)$, for $k \leq 2n$, we can partition the vector X into a number of disjoint sub-vectors (we say *components*):

$$\begin{array}{rcl} X_1 & = & (x_1, x_2, \ldots, x_{k_1}), \\ X_2 & = & (x_{k_1+1}, x_{k_1+2}, \ldots, x_{k_2}), \\ \ldots & = & \ldots \\ X_s & = & (x_{k_{s-1}+1}, x_{k_{s-1}+2}, \ldots, x_{k_s}), \end{array}$$

for some $s \in \mathbb{N}$ and $1 \leq k_1 < k_2 < \cdots < k_s = k$ in such a way that after placing one additional brush at v_1 (that is, after increasing x_1 by one), the process cleans k_1 vertices of the first component $(v_1, v_2, \ldots, v_{k_1})$; after placing two brushes, one at v_1 and one at v_{k_1+1}, the process cleans the first k_2 vertices; etc. Finally, when $s-1$ brushes are evenly distributed at the first coordinate of every component but the last one (that is, at vertices $v_1, v_{k_1+1}, v_{k_2+1}, \ldots, v_{k_{s-2}+1}$), the process cleans k_{s-1} vertices leaving us with

$$(k_{s-1} + x_{k_{s-1}+1}, k_{s-1} + x_{k_{s-1}+2}, \ldots, k_{s-1} + x_{k_s}).$$

We call the first $s-1$ components *dangerous*. If adding one more brush to the first coordinate of the last component (that is, to $v_{k_{s-1}+1}$) cleans the whole graph, we call the last component *dangerous* as well; otherwise we call it *safe*. Note that at the beginning of the game there is just one component. Moreover, this component is safe.

In order to prove the lemma, we introduce the strategy for the second player which makes sure that at the time when the first player is going to make her move, the corresponding vector X has the following properties:

(i) there is an even number of dangerous components,
(ii) each component has an even number of vertices,
(iii) each component has an even number of brushes,
(iv) safe component (if there is one) has the following property: $x_{2i+1} = x_{2i+2}$ for $k_{s-1}/2 \le i < k/2$ (that is, in the safe component each number of brushes occurs an even number of times).

We call such vectors *good*.

It is clear that the vector $(0, 0, \ldots, 0)$ corresponding to the first round of the game is good. Moreover, the strategy we described is a winning strategy: there is no chance to finish the game (in one round) from any non-trivial good vector X.

It remains to show that no matter what the first player does when playing on a good vector X, the second player can respond to get another good vector. We show this by case analysis.

Case 1: The first player plays on safe component by increasing x_{2i+1} (for some $i \in \mathbb{Z}, k_{s-1}/2 \le i < k/2$) but this component remains safe. The second player increases x_{2i+2} leaving the component in a safe state.

Case 2: The first player plays on safe component by increasing $x_{k_{s-1}+1}$ and as a result we decompose the safe component into a number of new dangerous components and, possibly, a smaller safe component. Note that increasing the maximum number of brushes in the safe component is the only chance to change the state of this component. It follows from (iv) that each new component has an even number of vertices. Moreover, all of them but the one containing $v_{k_{s-1}+1}$ has even number of brushes. The safe component (if there is any) still has (iv).

If the number of dangerous components is even, the second player adds one more brush to $v_{k_{s-1}+2}$ correcting the parity of brushes in the component. If the number of dangerous components is odd, the second player adds one more brush to $v_{k_{s-1}+1}$. After this move two things can happen. If there was a dangerous component before this round, we merge two components together and the vector is good. Otherwise, we clean all vertices in the first component and the vector becomes good too.

Case 3: The first player plays on dangerous component but the number of components remains the same. From (ii) and (iii) it follows that the second player can also play on this component not affecting the number of components.

Case 4: The first player plays on dangerous component merging this component with the other one. The second player puts one brush on the largest value at this new component. If it is the first component, then the component is cleaned after this move; otherwise it is merged to the previous one.

Case 5: The first player cleans the first dangerous component. The second player cleans the next one. □

7. Cleaning complete bipartite graphs

In this section, we try to analyze the complete bipartite graph $K_{n,m}$ with bipartite sets of size n and m, respectively. The case when both n and m are even is easy and follows immediately from Theorem 5.

Corollary 12. POLISH *played on* $K_{2n,2m}$ *is a second player win for any integers* $m \geq n \geq 1$.

We managed to analyze the game played on $K_{2,n}$, $n \in \mathbb{N}$ but all other graphs remain to be investigated. However, using a computer, we verified that $K_{2,3}$ is the only first player win complete bipartite graph $K_{n,m}$ with (a) $2 \leq n \leq m \leq 9$, (b) $2 \leq n \leq 7 \leq m \leq 12$, or (c) $2 \leq n \leq 6 \leq m \leq 15$. We conjecture that this is the case for the whole family of complete bipartite graphs with minimum degree at least 2.

Let us investigate the class of complete graphs $K_{2,n}$ for $n \in \mathbb{N}$. It is obvious that the first player wins the game played on $K_{2,1} = P_3$ and the second player wins on $K_{2,2} = C_4$ (see also previous lemma). It is also not so difficult to show that the first player wins the game on $K_{2,3}$. (The first player puts a brush on a vertex v_1 of degree 2. If the second player cleans v_1, she can clean the rest of the graph in the next round. If he puts a brush on a vertex of degree 3, then she responses by cleaning v_1 and the rest of the graph. So he has to put a brush on another vertex v_2 of degree 2. She does the same using the last vertex of degree 2 with no brush. The second player has to give up.) We will show that for all other cases the second player wins.

Theorem 13. POLISH *on* $K_{2,n}$ *is the second player win for any* $n \in \mathbb{N} \setminus \{1, 3\}$.

Proof. We discussed $n \leq 3$ already. The case when n is even follows from more general result, Theorem 5 (see also Corollary 12). It remains to consider $n = 2k + 1$ for some integer $k \geq 2$. Let us call each vertex in the bipartite set of size 2 a *center*; every other vertex will be called a *leaf*.

Suppose that the first player starts the game by placing a brush on a center. The second player can put a brush on another center to get the desired property he used to have for n even (see Theorem 5 and Corollary 12): the number of brushes on any vertex has the same parity as the number of dirty neighbours adjacent to it. He can clearly win from this position. Hence, the first player has to start with placing a brush on a leaf, and the second player responses in exactly the same way. After that the leaf is cleaned and each center receives one brush. We can repeat the same argument to show that she has to play on a leaf again. The second player keeps playing as before for a while but has to change his strategy after a certain number of rounds.

We need to consider two cases, depending on the parity of k. Suppose first that k is even. The second player forces the first one to play on leaves for the first $(k-1)$ rounds.

After that each center has $(k-1)$ brushes and there are $(2k+1)-(k-1)=(k+2)$ leaves. She still must play on a leaf but this time he responses by putting a brush on another leaf. If she keeps playing on leaves with no brush, he does the same (note the the number of leaves, $k+2$, is even which guarantees that it is possible). At some point she has to either clean a leaf or put a brush on a center. If she cleans a leaf (each center gets extra brush for a total of k brushes; the number of leaves drops to $k+1$), he plays on a center and wins. If she plays on a center, he cleans a leaf and wins again.

Suppose now that k is odd. This time, players play on leaves, cleaning one leaf per round, during the first $(k-2)$ rounds. After that each center has $(k-2)$ brushes and there are $(2k+1)-(k-2)=(k+3)$ leaves (again, even number). She still has to play on a leaf but he responses by putting a brush on a center: hence, one center has $(k-1)$ brushes, the other has $(k-2)$ brushes; there are $(k+3)$ leaves, exactly one with a brush. We need to consider 4 sub-cases.

Case 1: She cleans the only leaf with a brush. He responses by placing a brush on a center such that each center has k brushes; there are $(k+2)$ leaves, no leaf has a brush. She cannot clean anything in the next round but he can, regardless of her choice. She loses the game.

Case 2: She puts a brush on a center such that each center has $(k-1)$ brushes. This time he cleans the leaf and after this move the situation is exactly the same as before.

Case 3: She puts a brush on another center such that one center has k brushes and the other one $(k-2)$. The players keep playing on leaves with no brush (unless she increases the number of brushes on a center from $(k-2)$ to $(k-1)$; his response is to repeat the same move such that both centers have k brushes). At some point she has to either clean a leaf or increase the number of brushes on some center to $(k+1)$. Either way, she loses in the same round.

Case 4: She puts a brush on a leaf with no brush. He puts a brush on a center with $(k-2)$ brushes so that each center has $(k-1)$ brushes; there are exactly two leaves with a brush. If she cleans a leaf, he does the same and the game ends. If she puts a brush on a leaf with no brush, he does the same (recall that the number of leaves is even so it is always possible). If she plays on a center increasing the number of brushes to k, he does the same so that each center has k brushes. Thus, at some point she has to clean a leaf or increase the number of brushes on a center to $(k+1)$. Either way she loses, and the proof is complete. $\square$

8. LIGHT POLISH

Let G be a graph with brushes distributed on vertices in such a way that every vertex has discrepancy at most 3. We call this version LIGHT POLISH[2]. Suppose G is the disjoint union of $K_{1,3}$, $K_{2,3}$ (both with no brushes) and $K_{4,3}$ (with one brush on every vertex). Who wins and what is the winning move, if there is one? To answer this question we need to progress beyond outcome classes and evaluate the nim-values. This seems impractical for the full cleaning game but seems tractable for LIGHT POLISH.

[2]This variant was inspired by the game of TRAFFIC LIGHTS by Alan Parr. TRAFFIC LIGHTS is played on a grid, vertices are originally red, then yellow and then green. A line of 3 similarly coloured vertices wins the game.

We consider LIGHT POLISH of $K_{m,n}$ where $m, n \geq 1$. We will denote a position by $([i,j,k],[p,q,r])$ where a triple $[x,y,z]$ will denote the number of vertices of one of the colour classes that has x vertices with discrepancy 1 (needing one brush to fire), y the number with discrepancy 2, and z the number having discrepancy 3. Because adding a brush to a vertex requiring just one brush to fire will cause that vertex to fire and that can cause other vertices to fire, we will use $([a:i,j,k][p,q,r])$ as an intermediate step where the number before the colon indicates how many vertices are primed. A move in this presentation of the game will be one of the following:

$$\begin{aligned}
([i,j,k],[p,q,r]) \;&\Rightarrow\; ([i,j+1,k-1],[p,q,r]) \text{ or } ([i,j,k],[p,q+1,r-1]);\\
&\Rightarrow\; ([i+1,j-1,k],[p,q,r]) \text{ or } ([i,j,k],[p+1,q-1,r]);\\
&\Rightarrow\; ([1:i-1,j,k],[p,q,r]) \Rightarrow ([i-1,j,k],[p+q:r,0,0]) \text{ or}\\
& ([i,j,k],[1:p-1,q,r]) \Rightarrow ([i+j:k,0,0],[p-1,q,r]).
\end{aligned}$$

In the latter move, adding a brush to a vertex that requires just one extra brush to fire will cause that vertex to fire. All vertices in the other colour class will receive a brush but also will be incident with one fewer dirty edge, thereby decreasing discrepancy by two. This may of course cause more vertices to fire. For example, in $([1,0,2],[0,2,1])$ the possible moves are to $([1,1,1],[0,2,1])$, $([1,0,2],[0,3,0])$, $([1,0,2],[1,1,1])$, and $([1:0,0,2],[0,2,1]) \Rightarrow ([0,0,2],[2:1,0,0]) \Rightarrow ([2,0,0],[1:1,0,0]) \Rightarrow ([2:0,0,0],[1,0,0]) \Rightarrow 0$.

In order to account for the dirty edges, there are relationships between the numbers in each triple. Except for the fact that when one of the triples is [0,0,0] then the game is over, we ignore these relationships. For example, $([9,9,9],[0,0,1])$ is a legal position in the general game but there is no arrangement of brushes on $K_{m,n}$ for any m and n that would give this position.

We begin with an obvious but useful lemma.

Lemma 14. *Given the position $([i,j,k],[p,q,r])$ with $i > 0$, if $p+q \geq 2$ or if $p+q = 1$ and $i+j \geq 2$, then next player can finish the game on the next move.*

Proof. It follows from Theorem 2 that if several vertices in one colour class are primed that we may fire these before firing any in the other colour class.

Consider the move to $([1:i-1,j,k],[p,q,r]) \Rightarrow ([i-1,j,k],[p+q:r,0,0])$. If $r = 0$, then the game is already over. If $p+q \geq 2$ then the subsequent firing produces $([i-1+j+k:0,0,0],[r,0,0])$. After the $i-1+j+k$ vertices have fired, there are no vertices left in the one colour class and so the game is over. If $p+q = 1$, then the subsequent position is $([i-1+j:k,0,0],[r,0,0])$. If $i-1+j \geq 1$ then, after these vertices are fired, we are left with $[k,0,0],[r:0,0,0]$ and either the game is over $(k=0)$ or after the final firings the second colour class is incident with no dirty edges and so the game is over. □

From a position $([i,j,k],[p,q,r])$ there are six possible moves. In the mex operation this would allow values of 0 through 5 to occur. However, only the values 0 through 3 actually occur. Restricting all of i through r to be between 0 and 4 we get the distribution of values as value 0 occurs 5%; value 1 occurs 50%; value 2 occurs 44%;

and value 3 occurs 1%. In fact, almost all options are of the form (d.i) and (d.ii) and so almost all options have value 1 or 2—see Theorem 15.

To reduce some of the cases in the proof of the values for $K_{m,n}$, we introduce a new notation for a LIGHT POLISH position. A set of positions will be designated by $([a, b, c], [d, e, f], (x, y, z))$ where each of a through f will be of the form $*$, g or g^+ where g is a non-negative integer and each of x, y, z will be an element of $\{\cdot, 0, 1\}$. The entry $*$ means any non-negative integer is permissible; g^+ means any integer greater or equal to g is permissible; a 0 entry means that the sum of the corresponding entries in the position must be even, a 1 means it must be odd; and a $\cdot$ indicates there is no parity condition. Thus $([1, *, 0], [0, 1^+, 2^+]), (-, 1, 0)$ would be shorthand for all those positions $([i, j, k], [p, q, r]$ where $i = 1$, $j \geq 0$, $k = 0$, $p = 0$, $q \geq 1$, $r \geq 2$, $j + q$ must be odd and $k + r$ is even.

Theorem 15. *Let $\omega = ([i, j, k], [p, q, r])$ be a position in the* LIGHT POLISH *of $K_{m,n}$ where we will re-order so that $[i, j, k]$ is lexicographically greater than $[p, q, r]$. Then, $\mathcal{G}(\omega) \in \{0, 1, 2, 3\}$. Specifically:*

a. *if ω is of the form $([0, *, *], [0, *, *], (\cdot, x, y))$ then:*
   i. *if $(x, y) = (0, 0)$ then $\mathcal{G}(\omega) = 0$;*
   ii. *if $(x, y) = (0, 1)$ then $\mathcal{G}(\omega) = 1$;*
   iii. *if $(x, y) = (1, 0)$ then $\mathcal{G}(\omega) = 0$ except if ω is of the form $([0, 1, 1^+], [0, 0, 1], (\cdot, 1, 0))$ or $([0, 3^+, *], [0, 0, 1^+]), (\cdot, 1, 0)$ when $\mathcal{G}(\omega) = 2$;*
   iv. *if $(x, y) = (1, 1)$ then $\mathcal{G}(\omega) = 2$ except if ω is of the form $([0, 1, 2^+], [0, 0, 1], (\cdot, 1, 1))$ or $([0, 3^+, *], [0, 0, 1^+], (\cdot, 1, 1))$ when $\mathcal{G}(\omega) = 1$.*

b. *if ω is of the form $([1^+, *, *], [0, 0, 1^+], (\cdot, x, \cdot))$ then:*
   i. *if $x = 0$ then $\mathcal{G}(\omega) = 0$ except if ω is of the form $([1, 0, 1^+], [0, 0, 2^+], (\cdot, 0, \cdot))$, or $([1, 0, 0], [0, 0, 1^+], (\cdot, 0, \cdot))$ when $\mathcal{G}(\omega) = 1$;*
   ii. *if $x = 1$ then $\mathcal{G}(\omega) = 3$.*

c. *if ω is of the form $([1^+, *, *], [0, 1^+, *], (\cdot, x, \cdot))$ then:*
   i. *if $x = 0$ then $\mathcal{G}(\omega) = 1$;*
   ii. *if $x = 1$ then $\mathcal{G}(\omega) = 2$.*

d. *if ω is of the form $([1^+, *, *], [1^+, *, *], (\cdot, x, \cdot))$ then:*
   i. *if $x = 0$ then $\mathcal{G}(\omega) = 1$ except if ω of the form $([1, 0, 1^+], [1, 0, 1^+], (\cdot, 0, \cdot))$ when $\mathcal{G}(\omega) = 0$*
   ii. *if $x = 1$ then $\mathcal{G}(\omega) = 2$.*

Proof. We prove the result by induction. Recall that if we claim that $\mathcal{G}(\omega) = n$ then we have to show: (a) there is no move to an option whose nim-value is also n; (b) for each non-negative integer $m < n$ we have to show there is an option with that nim-value. We give the options, in order, for adding a brush to a vertex with deficiency 1, 2 and then 3, designated by (1) , (2) and (3), if such moves are possible.

Case a.: ω is of the form $([0, *, *], [0, *, *], (\cdot, x, y))$.

i. $x = y = 0$, to prove $\mathcal{G}(\omega) = 0$.
   (2) $\rightarrow ([1, *, *], [0, *, *], (\cdot, 1, 0))$ and so the option is either of the form $([1, *, *], [0, 0, 1^+], (\cdot, 1, 0))$ which is case (b.ii) and nim-value 3 or is of the form $([1, *, *], [0, 1^+, *], (\cdot, 1, 0))$ which is case (c.ii) and nim-value 2;

(3) $\to ([0,*,*],[0,*,*],(\cdot,1,1))$ and which is case (a.iv) and nim-value is either 1 or 2.

ii. $x=0$, $y=1$, to prove $\mathcal{G}(\omega)=1$.

(2) $\to ([1,*,*],[0,*,*],(\cdot,1,1))$ and so the option is either of the form $([1,*,*],[0,0,1^+],(\cdot,1,1))$ which is case (b.ii) and nim-value 3 or is of the form $([1,*,*],[0,1^+,*],(\cdot,1,1))$ which is case (c.ii) and nim-value 2;

(3) $\to ([0,*,*],[0,*,*],(\cdot,0,0))$ and which is case (a.i) and nim-value 0. Moreover, since $y=1$, this move always exists.

iii. $x=1$, $y=0$, to prove $\mathcal{G}(\omega)=0$ unless ω is an exception.

Exceptions: If ω is of the form $([0,1,1^+],[0,0,1],(\cdot,1,0))$ then a (2) move gives an option of the form $([1,0,1^+],[0,0,1],(\cdot,0,0))$ which has nim-value 0 from case (b.i), and a (3) moves gives an option of the form $([0,*,*],[0,*,*],(\cdot,0,1))$ which has nim-value 1 from case (a.ii).

If ω is of the form $([0,3^+,*],[0,0,1^+]),(\cdot,1,0)$ then a (2) move gives an option of the form $([1,2^+,*],[0,0,1^+],(\cdot,0,0))$ which has nim-value 0 from case (b.i), and a (3) moves gives an option of the form $([0,*,*],[0,*,*],(\cdot,0,1))$ which has nim-value 1 from case (a.ii).

In both cases, both moves always exist so that $\mathcal{G}(\omega)=2$.

General: If ω is of the form $([0,1,0],[0,0,1^+],(\cdot,1,0))$

(2) $\to ([1,0,0],[0,0,1^+],(\cdot,0,0))$ and the option has nim-value 1, case (b.i);

(3) $\to ([0,1,0],[0,1,*],(\cdot,0,1))$ and the option has nim-value 1, case (a.ii).

If ω is of the form $([0,1,1^+],[0,0,2^+],(\cdot,1,0))$

(2) $\to ([1,0,1^+],[0,0,2^+],(\cdot,0,0))$ and the option has nim-value 1, case (b.i);

(3) $\to ([0,*,*],[0,*,*],(\cdot,0,1))$ and the option has nim-value 1, case (a.ii).

The only sub-case remaining is when ω is of the form $([0,1^+,*],[0,1^+,*],(\cdot,1,0))$.

(2) $\to ([1,*,*],[0,1^+,*],(\cdot,0,0))$ and the option has nim-value 1, case (c.i);

(3) $\to ([0,*,*],[0,*,*],(\cdot,0,1))$ and the option has nim-value 1, case (a.ii).

In all cases, there is no option with nim-value 0.

iv. $x=1$, $y=1$, to prove $\mathcal{G}(\omega)=2$ unless ω is an exception.

Exceptions: If ω is of the form $([0,1,2^+],[0,0,1],(\cdot,1,1))$ then there are moves to $([1,0,1^+],[0,0,1],(\cdot,0,1))$ and to $([0,*,*],[0,*,*],(\cdot,0,0))$ both of which have nim-value 0 (b.i and a.i).

If ω is of the form $([0,3^+,*],[0,0,1^+],(\cdot,1,1))$ then there are moves to $([1,2^+,*],[0,0,1^+],(\cdot,0,1))$ and to $([0,*,*],[0,*,*],(\cdot,0,0))$ both of which have nim-value 0 (b.i and a.i).

Hence the exceptions have nim-value 1.

General: If ω is of the form $([0,1^+,*],[0,0,1^+],(\cdot,1,1))$ then the exceptions leave only the case $([0,1,0],[0,0,1],(\cdot,1,1))$. This has the options $([1,0,0],[0,0,1],(\cdot,0,1))$ and $([0,1,0],[0,1,0],(\cdot,0,0))$ which have nim-values 1 (b.i) and 0 (a.i) respectively and so the nim-value of ω is 2. Hence we are left with ω being of the form $([0,1^+,*],[0,1^+,*],(\cdot,1,1))$. This has options $([1,*,*],[0,1^+,*],(\cdot,0,1))$ and $([0,*,*],[0,*,*],(\cdot,0,0))$ which have nim-values

1 (c.i) and 0 (a.i) respectively. Hence the nim-value of ω is 2 since both moves always exist.

Case b.: ω is of the form $([1^+,*,*],[0,0,1^+],(\cdot,x,\cdot))$.

i. $x=0$, to prove $\mathcal{G}(\omega)=0$ unless ω is an exception.

Exceptions: Suppose ω is of the form $([1,0,1^+],[0,0,2^+],(\cdot,0,\cdot))$. A (1) move leaves $([2^+,0,0],[0,0,1^+],(\cdot,0,\cdot))$ which has nim-value 0, case (b.i). The (3) moves leave $([1,1,*],[0,0,2^+]),(\cdot,1,\cdot)$, nim-value 0 (b.ii) or $([1,0,1^+],[0,1,1^+]),(\cdot,1,\cdot)$, nim-value 2 (c.ii). All moves exists so $\mathcal{G}(\omega)=1$.

Suppose ω is of the form $([1,0,0],[0,0,1^+]),(\cdot,0,\cdot)$. A (1) move, which always exists, cleans the graph which has nim-value 0. The (3) moves leave $([1,0,0],[0,1,*]),(\cdot,1,\cdot)$ which has nim-value 2 (c.ii). Hence, $\mathcal{G}(\omega)=1$.

General: If ω is of the form $([1,0,1^+],[0,0,1],(\cdot,0,\cdot))$ then it has options $([1,0,0],[0,0,1^+],(\cdot,0,\cdot))$ and $([1,*,*],[0,*,*],(\cdot,1,\cdot))$ which have nim-values 1 (b.i) and 2 (c.ii) or 3 (b.ii) respectively. Hence ω has nim-value 0. Otherwise, ω is of the form $([1,1^+,*],[0,0,1^+],(\cdot,0,\cdot)$ or $([2^+,*,*],[0,0,1^+]),(\cdot,0,\cdot)$.

(1) $\to$ $([1^+,0,0],[0,1^+,*],(\cdot,0,\cdot)$ or $([1^+,*,*],[1^+,0,0]),(\cdot,0,\cdot)$ and both have nim-value 1 (c.i and d.i).

(2) $\to$ $([2^+,*,*],[0,0,1^+],(\cdot,1,\cdot)$ which has nim-value 3 (b.ii)

(3) $\to$ $([1^+,*,*],[0,*,*],(\cdot,1,\cdot)$ which has nim-value 2 or 3 (c.ii or b.ii).

Hence, $\mathcal{G}(\omega)=0$.

ii. $x=1$, ω is of the form $([1^+,*,*],[0,0,1^+],(\cdot,1,\cdot))$, to prove $\mathcal{G}(\omega)=3$. Note that the parity condition forces the form $([1^+,1^+,*],[0,0,1^+],(\cdot,1,\cdot))$.

(1) $\to$ $([1^+,0,0],[0,1^+,*],(\cdot,1,\cdot))$ or $([1^+,1^+,*],[1^+,0,0],(\cdot,1,\cdot))$ has nim-value 2 (c.ii or d.ii). One of these options always exists.

(2) $\to$ $([2^+,*,*],[0,0,1^+],(\cdot,0,\cdot))$ which has nim-value 0 (b.i). This move always exists.

(3) $\to$ $([1^+,1^+,*],[0,1,*],(\cdot,0,\cdot))$ which has nim-value 1 (c.i) and this option always exists, or $\to$ $([1^+,2^+,*],[0,0,1^+],(\cdot,0,\cdot))$ which has nim-value 0 (b.i)

Hence $\mathcal{G}(\omega)=3$.

Case c.: ω is of the form $([1^+,*,*],[0,1^+,*],(\cdot,x,\cdot))$.

i. $x=0$, to prove $\mathcal{G}(\omega)=1$.

(1) Since the number of vertices with deficiency 2 is at least two then this move will always clean the graph. This move, which always exists, gives an option with nim-value 0.

(2) $\to$ $([2^+,*,*],[0,1^+,*],(\cdot,1,\cdot))$ or $([1^+,*,*],[1,*,*],(\cdot,1,\cdot))$ and both have nim-value 2 (c.ii and d.ii).

(3) $\to$ $([1^+,*,*],[0,1^+,*],(\cdot,1,\cdot))$ which has nim-value 2 (c.ii).

Hence $\mathcal{G}(\omega)=1$.

ii. $x=1$, to prove $\mathcal{G}(\omega)=2$.

(1) This move will clean the graph, nim-value 0, except if ω is of the form $([1,0,1^+],[0,1,1^+],(\cdot,1,\cdot))$ when the option will be $([1^+,0,0],[1^+,0,0],(\cdot,0,\cdot))$ which has nim-value 1 (d.i).

(2) $\to$ $([2^+,*,*],[0,1^+,*],(\cdot,0,\cdot))$ or $([1^+,*,*],[1,*,*],(\cdot,0,\cdot))$ both of which have nim-value 1 (c.i and d.i) except if ω is of the form

$([1, 0, 1^+], [0, 1, 1^+], (\cdot, 1, \cdot))$ when the option will be $([1, 0, 1^+], [1, 0, 1^+], (\cdot, 0, \cdot))$ which has nim-value 0 (d.i).

(3) $\to ([1^+, *, *], [0, 1^+, *], (\cdot, 0, \cdot))$ which has nim-value 1 (c.i).

Hence $\mathcal{G}(\omega) = 2$ since moves (1) and (2) and then options of values 0 and 1 are always possible.

Case d.: ω is of the form $([1^+, *, *], [1^+, *, *], (\cdot, x, \cdot))$.

i. $x = 0$, to prove $\mathcal{G}(\omega) = 1$ unless ω is an exception.

(1) This move will clean the graph, nim-value 0, except if ω is of the form $([1, 0, 1^+], [1, 0, 1^+], (\cdot, 0, \cdot))$ when the option will be of the form $([1^+, 0, 0], [1^+, 0, 0], (\cdot, 0, \cdot))$ which has nim-value 1 (d.i). This move always exists.

(2) $\to ([2^+, *, *], [1^+, *, *], (\cdot, 1, \cdot))$ which has nim-value 2 (d.ii).

(3) $\to ([1^+, *, *], [1^+, *, *], (\cdot, 1, \cdot))$ which has nim-value 2 (d.ii).

Hence $\mathcal{G}(\omega) = 1$ except if ω is of the form $([1, 0, 1^+], [1, 0, 1^+], (\cdot, 0, \cdot))$ when $\mathcal{G}(\omega) = 0$.

ii. $x = 1$, to prove $\mathcal{G}(\omega) = 2$.

(1) This move, which always exists, will clean the graph, nim-value 0.

(2) $\to ([2^+, *, *], [1^+, *, *], (\cdot, 0, \cdot))$ which has nim-value 1 (d.i). The parity condition shows that this move always exists.

(3) $\to ([1^+, *, *], [1^+, *, *], (\cdot, 0, \cdot))$ which has nim-value 1 (d.i). (The parity condition eliminates the case that the option is $([1, 0, 1^+], [1, 0, 1^+], (\cdot, 0, \cdot))$.)

Hence $\mathcal{G}(\omega) = 2$.

This concludes the proof. □

9. Solution to the problem.

The problem was to find the winning move if G be the disjoint union of $K_{1,3}$, C_4, K_2 and $K_{2,3}$ and no brushes have yet been placed. Now $\mathcal{G}(K_{1,3}) = 0$, Theorem 8 (position is $(3, 3)$); $\mathcal{G}(C_4) = 0$, Theorem 5 or Lemma 6; $\mathcal{G}(K_2) = 1$, Lemma 6; and $\mathcal{G}(K_{2,3}) = 2$, Theorem 15 (position is $[0, 3, 0], [0, 0, 2]$); so that

$$\begin{aligned} \mathcal{G}(G) &= \mathcal{G}(K_{1,3}) \oplus \mathcal{G}(C_4) \oplus \mathcal{G}(K_2) \oplus \mathcal{G}(K_{2,3}) \\ &= 0 \oplus 0 \oplus 1 \oplus 2 \\ &= 3. \end{aligned}$$

Consider the move of placing a brush on a vertex of degree 2 in $K_{2,3}$ giving the position $[0, 3, 0], [0, 1, 1]$. This has nim-value 1 and the total game has nim-value $0 \oplus 0 \oplus 1 \oplus 1 = 0$ thus this is a winning move. Showing that there is no other winning move we leave to the reader.

References

[1] M.H. Albert, R.J. Nowakowski, and D. Wolfe, *Lessons in Play*, A K Peters, Ltd., 2007.

[2] N. Alon, P. Prałat, and N. Wormald, Cleaning regular graphs with brushes, *SIAM Journal on Discrete Mathematics* **23** (2008), 233–250.

[3] E.R. Berlekamp, J.H. Conway, and R.K. Guy, *Winning ways for your mathematical plays*, Vol. 1 A K Peters, Ltd., 2001.

[4] T. Biedl, T. Chan, Y. Ganjali, M. Hajiaghayi, and D. Wood, Balanced vertex–orderings of graphs, *Discrete Applied Mathematics* **148**(1) (2005) 27–48.
[5] A. Bjorner, L. Lovasz, and P.W. Shor, Chip-firing games on graphs, *European J. Combin.*, **12** (1991), 283–291.
[6] J.H. Conway, *On Numbers and Games*, 2nd Edition, A K Peters, Ltd., 2001.
[7] F.V. Fomin and D.M. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science*, **399** (2008) 236–245.
[8] S. Gaspers, M.-E. Messinger, R. Nowakowski, and P. Prałat, Clean the graph before you draw it!, *Information Processing Letters* **109** (2009), 463–467.
[9] S. Gaspers, M.-E. Messinger, R. Nowakowski, and P. Prałat, Parallel cleaning of a network with brushes, *Discrete Applied Mathematics* **158** (2009) 467–478.
[10] P. Gordinowicz, R. Nowakowski, P. Prałat, and A. Siegel, French polish, preprint.
[11] A. Guo and E. Miller, Lattice point methods for combinatorial games, *Advances in Applied Mathematics*, **46** (2011) 363–378.
[12] B. Hobbs and J. Kahabka, Underwater Cleaning Technique Used for Removal of Zebra Mussels at the Fitzpatrick Nuclear Power Plan, *Proceedings of The Fifth International Zebra Mussel and Other Aquatic Nuisance Organisms Conference, The Sea Grant Nonindigenous Species Site*. 1995. accessed: 4 June, 2008 <http://www.sgnis.org/publicat/47.htm>
[13] J. Kára, J. Kratochvíl, and D. Wood, On the Complexity of the Balanced Vertex Ordering Problem, *Discrete Mathematics & Theoretical Computer Science* **9** (1) (2007) 193–202.
[14] S.R. Kotler, E.C. Mallen, K.M. Tammus, Robotic Removal of Zebra Mussel Accumulations in a Nuclear Power Plant Screenhouse, *Proceedings of The Fifth International Zebra Mussel and Other Aquatic Nuisance Organisms Conference, The Sea Grant Nonindigenous Species Site*. 1995. accessed: 4 June 2008 <http://www.sgnis.org/publicat/113.htm>
[15] M.-E. Messinger, R. J. Nowakowski, and P. Prałat, Cleaning a network with brushes, *Theoretical Computer Science* **399** (2008), 191–205.
[16] M.E. Messinger, R. Nowakowski, and P. Prałat, Cleaning with Brooms, *Graphs and Combinatorics* **27** (2011), 251–267.
[17] M.-E. Messinger, R. J. Nowakowski, P. Prałat, and N. Wormald, Cleaning random d–regular graphs with brushes using a degree-greedy algorithm, *Proceedings of the 4th Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN 2007)*, Lecture Notes in Computer Science **4852**, Springer, 2007, 13–26.
[18] P. Prałat, Cleaning random d–regular graphs with Brooms, *Graphs and Combinatorics* **27** (2011), 567–584.
[19] P. Prałat, Cleaning random graphs with brushes, *Australasian Journal of Combinatorics* **43** (2009), 237–251.
[20] D. Singmaster, Almost all games are first person games, *E*ureka, **41** (1981) 33–37.

Department of Mathematics, Technical University of Lodz, Łódź, Poland
E-mail address: pgordin@p.lodz.pl

Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada B3H 3J5
E-mail address: rjn@mathstat.dal.ca

Department of Mathematics, Ryerson University, Toronto, ON, Canada, M5B 2K3
E-mail address: pralat@ryerson.ca