SIAM J. DISCRETE MATH.
Vol. 21, No. 4, pp. 999–1018

COLORING BULL-FREE PERFECTLY CONTRACTILE GRAPHS*

BENJAMIN LÉVÊQUE† AND FRÉDÉRIC MAFFRAY‡

Abstract. We consider the class of graphs that contain no bull, no odd hole, and no antihole of length at least five. We present a new algorithm that colors optimally the vertices of every graph in this class. This algorithm is based on the existence in every such graph of an ordering of the vertices with a special property. More generally we prove, using a variant of lexicographic breadth-first search, that in every graph that contains no bull and no hole of length at least five there is a vertex that is not the middle of a chordless path on five vertices. This latter fact also generalizes known results about chordal bipartite graphs, totally balanced matrices, and strongly chordal graphs.

Key words. perfect graph, bull-free graph, coloring, LexBFS

AMS subject classifications. 05C15, 05C17, 05C85

DOI. 10.1137/06065948X

1. Introduction.

The chromatic number of a graph G is the smallest integer $\chi(G)$ for which it is possible to assign one color from the set $\{1, \dots, \chi(G)\}$ to each vertex so that any two adjacent vertices receive different colors. A graph G is *perfect* if the chromatic number of every induced subgraph H of G is equal to $\omega(H)$, where $\omega(H)$ is the maximum clique size in H. A *hole* is a chordless cycle on at least four vertices. The complement of a hole is called an *antihole*. A hole or an antihole is odd if it has an odd number of vertices. Graphs that do not contain an odd hole or an odd antihole of length at least five are usually called Berge graphs. Berge [2, 3] conjectured that such graphs are perfect, and this famous problem, known as the strong perfect graph conjecture, was solved by Chudnovsky et al. [5]. Earlier, Grötschel, Lovász, and Schrijver [15] gave a polynomial time algorithm that computes the chromatic number of every perfect graph; but this algorithm, based on the ellipsoid method, is considered very impractical, and it is still an open problem to find a purely combinatorial algorithm to color optimally the vertices of all perfect graphs in polynomial time. Here we consider the class of bull-free graphs.

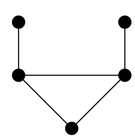

FIG. 1. *The bull.*

A *bull* is a graph with five vertices a, b, c, d, e and edges ab, bc, cd, de, bd; see Figure 1. We will denote such a bull by *a-bcd-e*. In a bull *a-bcd-e*, we call the edge bd the *central* edge and vertices b, d the *ears* of the bull. Chvátal and Sbihi [8] proved that

*Received by the editors May 10, 2006; accepted for publication (in revised form) May 11, 2007; published electronically January 9, 2008. This research was supported by the ADONET network of the European Community, which is a Marie Curie Training Network.
http://www.siam.org/journals/sidma/21-4/65948.html
†Laboratoire G-SCOP, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France (benjamin.leveque@g-scop.inpg.fr).
‡C.N.R.S., Laboratoire G-SCOP, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France (frederic.maffray@g-scop.inpg.fr).

the strong perfect graph conjecture holds for bull-free graphs, that is, every bull-free Berge graph is perfect. Subsequently, the structure of bull-free Berge graphs was also studied by Reed and Sbihi [31]; De Figueiredo, Maffray, and Porto [10, 11]; and Hayward [18]. De Figueiredo and Maffray [9] gave a combinatorial algorithm, based on the results from [8, 10], that optimally colors every bull-free Berge graph G with n vertices and m edges in time $\mathcal{O}(n^5m^3)$.

Let $\mathcal{B}$ be the class of bull-free Berge graphs that contain no antihole of length at least five. We will present an $\mathcal{O}(mn)$ algorithm that computes an optimal coloring for every graph in class $\mathcal{B}$. This algorithm is based on new structural results concerning the graphs in that class. Before doing so, we want to review the known methods that perform such a task, and for this purpose we need to introduce a few more definitions.

A graph G is *weakly chordal* [17] if G contains no hole of length at least five and no antihole of length at least five. A graph G is *transitively orientable* [14, 28] if we can assign one orientation to each of its edges so that for every directed path $u \rightarrow v \rightarrow w$ the arc $u \rightarrow w$ is present in the orientation. A graph G is *perfectly orderable* [6] if it admits an ordering $<$ such that, for every induced subgraph H of G, applying the greedy coloring algorithm on $(H, <)$ produces an optimal coloring (such an ordering is called a perfect ordering). A *homogeneous set* in a graph G is a set $S \subset V(G)$ with $|S| \geq 2$, $S \neq V(G)$, such that every vertex of $V(G) \setminus S$ is adjacent to either all or none of the vertices of S. A *prism* is a graph that consists in two disjoint triangles and three disjoint paths between the two triangles, with no edge between any two of these three paths other than the triangles' edges. A prism is odd if these three paths have odd length. A graph G is an *Artemis* graph [12] if it contains no odd hole, no antihole of length at least five, and no prism. A graph G is a *Grenoble* graph [12] if it contains no odd hole, no antihole of length at least five, and no odd prism. It was proved in [10] that every graph in class $\mathcal{B}$ is "perfectly contractile" in the sense of Bertschi [4]; see section 5. Note that a prism either is the complement of a cycle of length six or contains a bull. Therefore, "bull-free Artemis," "bull-free Grenoble," and "bull-free perfectly contractile" are just different names for class $\mathcal{B}$.

We know of three purely combinatorial methods to color graphs in class $\mathcal{B}$, which we summarize briefly:

• Method 1: Results from [10, 11] say that every graph in class $\mathcal{B}$ either is weakly chordal, or has a homogeneous set, or is transitively orientable. Homogeneous sets can be handled by the so-called *modular decomposition*, which decomposes any graph into $\mathcal{O}(n)$ subgraphs that have no homogeneous sets. Modular decomposition can be performed in time $\mathcal{O}(n + m)$; see, for example, [16]. By [10, 11], for a graph in class $\mathcal{B}$, these indecomposable subgraphs are either weakly chordal or transitively orientable. One can find an optimal coloring for these subgraphs in time $\mathcal{O}(nm)$ for weakly chordal graphs [19] and in time $\mathcal{O}(m)$ for transitively orientable graphs [27]. One can then combine these optimal colorings along the modular decomposition to obtain an optimal coloring of the original graph (details are omitted). Thus we can estimate the complexity of this method at $\mathcal{O}(n^2m)$.

• Method 2: Chvátal [7] conjectured that every graph in class $\mathcal{B}$ is perfectly orderable, and Hayward [18] proved that conjecture, using some results from [10, 11]. We estimate the technique in [18] at $\mathcal{O}(n^5)$ (the exponent 5 is due to the search for an induced P_5 performed in [11]), and so, combining the techniques in [10, 11, 18], and using again a linear-time algorithm for modular decomposition such as [16], one can find a perfect ordering of any graph in class $\mathcal{B}$ in time $\mathcal{O}(n^5(n + m))$. Then applying the greedy coloring on this ordering produces an optimal coloring in time $\mathcal{O}(m)$. Thus the total complexity of this method can be estimated at $\mathcal{O}(n^5(n+m))$.

• Method 3: Since every graph in class $\mathcal{B}$ is an Artemis graph, one can use the algorithm from [25], which colors every Artemis graph in time $\mathcal{O}(n^2m)$.

Our aim here is to present an algorithm that we think is conceptually simpler than all of the above and whose complexity is also lower.

First let us fix some terminology and notation. We say that a vertex a *sees* a vertex b when ab is an edge of the graph, otherwise vertex a *misses* b. The complement of a graph G is denoted by $\overline{G}$. The neighborhood of a vertex v is denoted by $N(v)$. The degree of a vertex v in G is denoted by $d(v)$. A chordless path on k vertices is denoted by P_k. A *house* is a graph with five vertices a, b, c, d, e and edges ac, ce, eb, bd, da, ae; vertex c is called the *top* of the house. Note that a house is the complement of a P_5. We will establish the following result.

THEOREM 1.1. *Every graph in $\mathcal{B}$ has a vertex that is not the top of a house.*

The above theorem implies the following. Let G be any graph in $\mathcal{B}$. So G has a vertex v_1 that is not the top of a house, and for $i = 2, \dots, n$, the subgraph $G \setminus \{v_1, \dots, v_{i-1}\}$ has a vertex v_i that is not the top of a house in this subgraph. We may call the ordering $v_1, \dots, v_n$ of the vertices of G an *NTH elimination ordering.* In section 3 we show how such an ordering can be computed in time $\mathcal{O}(nm)$, using the algorithm described in section 2. After such an ordering is obtained, we run an $\mathcal{O}(nm)$ coloring algorithm called COSINE*, which is a new algorithm based on Hertz's coloring algorithm COSINE [21]. Algorithm COSINE works on a graph whose vertices need not be ordered, while COSINE* uses the NTH elimination ordering. In section 5 we prove the optimallity of this coloring algorithm for every graph in $\mathcal{B}$. In section 6 we present an extension of this algorithm that finds a clique of maximum size in a graph in $\mathcal{B}$. This yields an $\mathcal{O}(nm)$ robust algorithm to color graphs in $\mathcal{B}$.

Let $\mathcal{C}$ be the class of graphs that contain no bull and no hole of length at least five. Clearly $\mathcal{B}$ is strictly contained in $\overline{\mathcal{C}}$, and Theorem 1.1 is an immediate consequence of the following.

THEOREM 1.2. *Every graph in $\mathcal{C}$ has a vertex that is not the middle of a P_5.*

The above theorem will be proved in section 3. Note that this theorem implies the following. Let G be any graph in $\mathcal{C}$. So G has a vertex v_1 that is not the middle of a P_5, and for $i = 2, \dots, n$, the subgraph $G \setminus \{v_1, \dots, v_{i-1}\}$ has a vertex v_i that is not the middle of a P_5 in this subgraph. We may call the ordering $v_1, \dots, v_n$ of the vertices of G an *NMP$_5$ elimination ordering.* The proof of Theorem 1.2 is an $\mathcal{O}(nm)$ algorithm called LEXBFS* that finds such an ordering.

We mention a theoretical consequence of this theorem. Recall that a graph is *chordal bipartite* if it is bipartite and it contains no hole of length at least six. A classical result is the existence in every chordal bipartite graph of a vertex that is not the middle of a P_5. This result is known under several equivalent variants, such as the existence of a *simple* vertex in every *strongly chordal* graph, or the existence of a Γ-*free* ordering in every *totally balanced* matrix [26]. Since every chordal bipartite graph is in class $\mathcal{C}$, our Theorem 1.2 generalizes this result.

2. Algorithm LEXBFS*.

Algorithm LEXBFS* is a particular case of Algorithm LEXBFS (lexicographic breadth-first search). Algorithm LEXBFS, due to Rose, Tarjan, and Lueker [32], explores a graph and numbers its vertices one by one, from n to 1. At the general step, each unnumbered vertex has a label, which is the set of numbers of its already numbered neighbors. A lexicographic order is defined on the labels: label $L(a)$ is strictly greater than label $L(b)$ if there exists an integer i such that $i \in L(a) \setminus L(b)$ and $\forall j > i$, either $j \in L(a) \cap L(b)$ or $j \notin L(a) \cup L(b)$. The next vertex to be numbered is any unnumbered vertex whose label is lexicographically maximal. Ties in LexBFS are broken arbitrarily.

In LexBFS*, we need to break ties according to the following rule. Suppose that at a given step the set A of unnumbered vertices with maximal label satisfies $|A| \geq 2$. Let $L(A)$ be the label of the vertices in A. Let U be the set of unnumbered vertices not in A. For each $u \in U$, set $L'(u) := L(u) \setminus L(A)$, and let the vertices of U be ordered lexicographically according to L'. Then the first (i.e., maximal according to the L' ordering) vertex u of U "votes" by eliminating from A the nonneighbors of u (except if that causes A to become empty; in that case u has no effect); then the second vertex of U votes, etc. The procedure stops when all vertices of U have voted; then ties are broken arbitrarily. Here is a formal description of the algorithm:

Algorithm LexBFS*
Input: A graph G with n vertices.
Output: An ordering σ on the vertices of G.
Initialization: For every vertex a of G, set $L(a) := \emptyset$;
General step: For $i = n, \dots, 1$ do:
1. Let A be the set of unnumbered vertices whose label is maximum, and let U be the other unnumbered vertices.
2. While $U \neq \emptyset$ do:
   2.1. Select a vertex $u \in U$ for which $L(u) \setminus L(A)$ is maximum.
   2.2. Set $U := U \setminus \{u\}$. If $A \cap N(u) \neq \emptyset$, then set $A := A \cap N(u)$.
3. Pick any vertex $a \in A$ and set $\sigma(a) := i$.
4. For each unnumbered neighbor v of a, add i to $L(v)$.

Complexity analysis. Let us analyze the complexity of Algorithm LexBFS*. Rose, Tarjan, and Lueker [32] showed that Algorithm LexBFS can be implemented in time $\mathcal{O}(n+m)$ as follows, where n is the number of vertices and m the number of edges of the graph in input. Ordering the vertices according to the value of $L(v)$ can be done with the usual techniques, such as bucket sort [1]: For each label ℓ, we maintain the set S_ℓ of the unnumbered vertices v such that $L(v) = \ell$. This set is implemented as a doubly linked list, where each element also points to the head of the list, which is a special cell containing their label. The heads of the nonempty S_ℓ's are themselves put in decreasing lexicographic label order into a doubly linked list M. During the initialization step, all vertices are put into $S_\emptyset$, and $S_\emptyset$ is the only element of M. Thus the initialization takes time $\mathcal{O}(n)$. Set A of step 1 of the algorithm is the first set in M. When a vertex a of A is selected at step 3, it is removed from the data structure, and each neighbor u of a is removed from the set S_ℓ that contains u and added into a (new) set $S_{\ell \cup \{\sigma(a)\}} = S_\ell \cap N(A)$ which is placed just before S_ℓ in M (empty sets are removed from M). This operation of splitting the S_ℓ's takes time $\mathcal{O}(d(a))$. So the total cost of steps 3 and 4 is $\mathcal{O}(n+m)$. This is how LexBFS is implemented in [32].

Unfortunately, breaking the ties in LexBFS* increases the complexity to $\mathcal{O}(nm)$ as we show now. Consider the set U defined on line 1 of the algorithm. Set U is ordered according to $L'(u)$ by using the same data structure as before. This takes time $\mathcal{O}(n+m)$. This ordering procedure is performed only once, at the beginning of step 2. Then, at step 2.1 we take the maximum vertex u in the ordered set U (which takes constant time), and the operations performed in step 2.2 take time $\mathcal{O}(d(u))$. So the total cost of step 2 is $\mathcal{O}(n+m)$. Since this step is performed n times, the total running time of Algorithm LexBFS* is $\mathcal{O}(n(n+m))$.

Actually, we will need to apply Algorithm LexBFS* on the complement $\overline{G}$ of a graph G. Let $\overline{m}$ be the number of edges in $\overline{G}$. Since $\overline{m} = \mathcal{O}(n^2)$, this might lead to a complexity of $\mathcal{O}(n^3)$, but we can avoid this as follows. When applied on $\overline{G}$, splitting the sets S_ℓ take time $\mathcal{O}(\overline{d}(a))$, where $\overline{d}$ is the degree function in $\overline{G}$, but we can do it in time $\mathcal{O}(d(a))$ if, instead of removing each neighbor u of a (in $\overline{G}$) from the set S_ℓ that contains u and adding it into the new set $S_\ell \cap N_{\overline{G}}(A)$, we remove each neighbor u of a (in G) from the set S_ℓ that contains u and add it into a new set $S_\ell \setminus N_G(A)$, which is placed just after S_ℓ in M. The same idea can be used to sort the set U and to update A in time $\mathcal{O}(n+m)$. In conclusion, the total running time of Algorithm LexBFS* applied on the complement $\overline{G}$ of a graph G with n vertices and m edges is $O(nm)$.

Properties of LexBFS. Here are some notation and properties for Algorithm LexBFS. When the algorithm selects a vertex $a \in A$ at step 3 of Algorithm LexBFS, we denote by $L_a(u)$ the current value of the label of any vertex u at this step of the algorithm. We denote by $a < b$ the fact that $\sigma(a) < \sigma(b)$.

Lemma 2.1. *Suppose that $a < u$, $b \le u$, and $L_u(a) < L_u(b)$. Then $a < b$ and, $\forall v$ such that $v \le u$, $L_v(a) < L_v(b)$.*

Proof. Suppose $a < u$, $b \le u$, and $L_u(a) < L_u(b)$. At the step of the algorithm when u is numbered, there exists $i > \sigma(u)$ such that $i \in L_u(b) \setminus L_u(a)$ and $\forall j > i$, either $j \in L_u(a) \cap L_u(b)$ or $j \notin L_u(a) \cup L_u(b)$. After u is numbered, integers that may be added to $L(a)$ and $L(b)$ are smaller than $\sigma(u)$ and therefore strictly smaller than i, so the inequality $L(a) < L(b)$ still holds throughout the rest of the execution of the algorithm. Thus the lemma holds. ∎

Lemma 2.2. *Suppose that $a < b$ and $L_b(a) \neq L_b(b)$. Then there exists a vertex $> b$ that sees b and misses a. Let $f(b, a)$ be a maximum such vertex. Then we have the following properties:*

- *For every u that sees a and misses b, we have $u < f(b, a)$.*
- *Every u such that $f(b, a) < u$ either sees both a, b or misses both a, b.*

Proof. Suppose $a < b$ and $L_b(a) \neq L_b(b)$. Then $L_b(a) < L_b(b)$ because b is selected before a. Then there exists i such that $i \in L_b(b) \setminus L_b(a)$ and $\forall j > i$, either $j \in L_b(a) \cap L_b(b)$ or $j \notin L_b(a) \cup L_b(b)$. Vertex $f(b, a)$ is the vertex such that $\sigma(f(b, a)) = i$.

Suppose a vertex u sees a, misses b, and $u > f(b, a)$. Let $j = \sigma(u)$. Since u sees a, we have $j \in L_b(a)$. Since u misses b, we have $j \notin L_b(b)$. So $j \in L_b(a) \setminus L_b(b)$, a contradiction to the definition of i.

Let u' be a vertex such that $f(b, a) < u'$. Let $j' = \sigma(u')$. Since $j' = \sigma(u') > \sigma(f(b, a)) = i$, we have $j \in L_b(a) \cap L_b(b)$ or $j \notin L_b(a) \cup L_b(b)$, and so u' either sees both a, b or u' misses both. Thus the lemma holds. ∎

Lemma 2.3. *Suppose that $a < b < u$, and u sees a and misses b. Let $a_0 = a$, $b_0 = b$, $a_1 = u$, $b_1 = f(b, a)$, and define vertices a_i and b_i, for $i \ge 2$, as follows, as long as possible:*

- *If b_i misses a_i, then let $a_{i+1} = f(a_i, b_{i-1})$.*
- *If a_{i+1} misses b_i, then let $b_{i+1} = f(b_i, a_i)$.*

Let k be the maximum integer such that a_k is defined. Let ℓ be the maximum integer such that b_ℓ is defined, so ℓ is equal to k or $k + 1$. Denote by $\mathcal{P}(u, b, a)$ the path a_0-$\cdots$-a_k-b_ℓ-$\cdots$-b_0. If a misses b, then $\mathcal{P}(u, b, a)$ is a chordless path. If a sees b, then $\mathcal{P}(u, b, a)$ is a hole.

Proof. Suppose $\ell = k$ for convenience (the same can be done when $\ell = k + 1$). We prove by induction on $j \le k$ the property that the sequences $(a_i)_{i \le j}$, $(b_i)_{i \le j}$ are well defined, $a_0 < b_0 < a_1 < b_1 < \cdots < a_j < b_j$, a_0-$\cdots$-a_j and b_0-$\cdots$-b_j are chordless paths, and there is no edge between the (a_i)'s and the (b_i)'s, except for $a_k b_k$ and possibly a_0b_0.

If $j = 1$, then a_1 sees a_0, misses b_0, and $a_0 < b_0 < a_1$, so $L_{b_0}(a_0) \neq L_{b_0}(b_0)$. So vertex $b_1 = f(b_0, a_0)$ is well defined by Lemma 2.2. Vertex b_1 sees b_0, misses a_0, and $a_1 < b_1$. So the property is true for $j = 1$.

Now suppose that $1 \leq j < k$ and that the property is true for j. Since b_j sees b_{j-1}, misses a_j, and $b_{j-1} < a_j < b_j$, we have $L_{a_j}(b_{j-1}) \neq L_{a_j}(a_j)$. Apply Lemma 2.2 to define $a_{j+1} = f(a_j, b_{j-1})$. Vertex a_{j+1} sees a_j, misses b_{j-1}, and $b_j < a_{j+1}$. Since a_{j+1} misses b_{j-1}, and $a_0 < b_0 < a_1 < b_1 = f(b_0, a_0) \cdots < a_j = f(a_{j-1}, b_{j-2}) < b_j = f(b_{j-1}, a_{j-1})$, it follows that a_{j+1} misses $a_0, \ldots, a_{j-1}, b_0, \ldots, b_{j-1}$. The same can be done to define b_{j+1}. So the property is true for $j + 1$. Thus the lemma holds. □

LEMMA 2.4. *In a graph that contains no hole of length at least five, suppose that $a < b < u$, u sees a, u misses b, and a sees b. Then $f(b, a)$ sees u.*

Proof. Consider the path $\mathcal{P}(u, b, a)$ of Lemma 2.3. Since a sees b, that path is a hole, so it is a hole of length four, so $f(b, a)$ sees u. □

Properties of LEXBFS*. Here are some notation and properties for Algorithm LEXBFS*. When the algorithm selects a vertex $a \in A$ at step 3 of Algorithm LEXBFS*, we put $L'_a(u) = L_a(u) \setminus L_a(a)$ for every (unnumbered) vertex u.

LEMMA 2.5. *Suppose that $a < b$, $L_b(a) = L_b(b)$, and $N(a) \neq N(b)$. Then, during the loop of step 2 of algorithm* LEXBFS*, *vertex a has been removed from A by a vertex $u = g(b, a)$ that sees b and misses a. We have the following properties:*
- *$u < a$,*
- *$L_b(u) < L_b(b)$,*

if there exists a vertex $v < a$ that sees a, misses b, and $L_b(v) \neq L_b(b)$, then $L'_b(v) \leq L'_b(u)$. If $L'_b(v) \neq L'_b(u)$, then there exists a vertex $> b$ that sees u and misses a, b, v, denote by $x = h(u, v)$ a maximum such vertex. We have the following properties:
- *For all y that sees v and misses a, b, u, we have $y < x$.*
- *For all y such that $x < y$ and y misses a, b, we have y sees u, v or y misses u, v.*

Proof. The definition of u and its properties follows from the definition of the algorithm. Suppose there exists a vertex $v < a$ that sees a, misses b, and $L_b(v) \neq L_b(b)$.

Suppose that $L'_b(v) > L'_b(u)$. Then v should have been selected at step 2.1 before u. Then, at step 2.2, $A \cap N(v)$ should be empty, otherwise b is removed from A and b is not the selected vertex at step 3. Since a is in $N(v)$, it has been previously removed from A by a vertex w with $L'_b(w) \geq L'_b(v)$. Since $L'_b(w) \geq L'_b(v) > L'_b(u)$, so $w \neq u$. This contradicts the definition of $u = g(b, a)$, so $L'_b(v) \leq L'_b(u)$.

If $L'_b(v) \neq L'_b(u)$, then $x = h(u, v)$ is well defined.

Suppose there exists a vertex y that sees v, misses a, b, u, and $x < y$. Then $L'_b(v) < L'_b(u)$ implies that there exists a vertex $> y$ that sees u and misses a, b, v; a contradiction to the definition of x.

Let y' be a vertex such that $x < y'$ and y' misses a, b. By the preceding property, it is not possible that y' sees v and misses u. If y' sees u and misses v, then this is a contradiction to the definition of x. So y sees u, v or y misses u, v. Thus the lemma holds. □

3. Proof of Theorem 1.2.

Recall that $\mathcal{C}$ denotes the class of graphs that contain no bull and no hole of length at least five. In this section we prove that when the input graph is in $\mathcal{C}$, the ordering given by Algorithm LEXBFS* is an NMP_5 elimination ordering. It may be worth pointing out that this outcome does not hold for

LexBFS. For an example, consider the graph made of a chordless path a-b-c-d-e-f-g plus one vertex h adjacent to a, c, e, g. Then LexBFS can produce the ordering h, a, g, c, e, b, f, d, and d is the middle of the P_5 b-c-d-e-f. It is this example that led us to define the tie-breaking rule of LexBFS*.

Before proving the main result, we need the following lemma.

Lemma 3.1. *In a graph $G \in \mathcal{C}$, let $P = a_0$-a_1-$\cdots$-a_r be a chordless path with $r \geq 4$, and let u be a vertex that sees the two endvertices a_0, a_r of P. Then one of the following holds*:

- *u sees all vertices of P,*
- *r is even, and u sees $a_0, a_2, \ldots, a_r$ and misses $a_1, a_3, \ldots, a_{r-1}$, or*
- *$r = 4$, and u sees a_2 and exactly one of a_1, a_3.*

Consequently, in any case, u sees a_2 and a_{r-2}.

Proof. Denote a *segment* as any subpath of P, of length at least one, whose endvertices see u and interior vertices do not. So P is (edgewise) partitioned into its segments. Since G contains no hole of length at least five, every segment has length one or two. For $\ell = 1, 2$, let s_ℓ be the number of segments of P of length ℓ. So $r = s_1 + 2s_2$. If $s_1 = 0$, then every segment has length two, and we have the second outcome of the lemma. Now let $s_1 > 0$. So u sees two consecutive vertices of P. Suppose that we do not have the first outcome, so u has a nonneighbor in P. Thus, up to symmetry, there is an integer i such that u sees a_i and a_{i+1} and not a_{i+2}. Then $i \leq 1$, for otherwise a_0-ua_ia_{i+1}-a_{i+2} is a bull, and $r \leq i + 3$, for otherwise a_r-ua_ia_{i+1}-a_{i+2} is a bull. It follows that $r = 4$ and $i = 1$, and we have the third outcome. Thus the lemma holds. □

Now we prove the following theorem, which implies Theorem 1.2. For any path P, let P^* denote the path formed by the interior vertices of P.

Theorem 3.2. *When the input graph is a graph in $\mathcal{C}$, Algorithm* LexBFS* *produces an NMP$_5$ ordering of the vertices of G.*

Proof of Theorem 3.2. Say that a P_5 a-b-c-d-e in G is *bad* if $c < \min\{a, b, d, e\}$. Say that a bad P_5 a-b-c-d-e is *worse* than another bad P_5 a'-b'-c'-d'-e' if $a \geq a', b \geq b', c \geq c', d \geq d', e \geq e'$, and at least one of these five inequalities is strict. Our aim is to prove that there is no bad P_5, so let us assume the contrary and show that this leads to a contradiction. Let a-b-c-d-e be a worst P_5. Up to symmetry we may assume that $e < a$.

Claim 1. $e < b$.

Proof. Suppose the claim is false, so $c < b < e < a$.

Since a sees b, misses e, and $b < e < a$, we can consider the chordless path $R = \mathcal{P}(a, e, b)$ of Lemma 2.3. If none of c, d has a neighbor in R^*, then $R \cup \{c, d\}$ is a cycle of length at least six, so one of c, d has a neighbor in R^*. Let q be the vertex of R^* closest to a that sees one of c, d. If q misses c, then $R[b, q] \cup \{d, c\}$ is a hole of length ≥ 5. So q sees c. The hole $R[b, q] \cup \{c\}$ must have length < 5, so q sees a and so $q \neq d$.

Since q sees c, misses b, and $c < b < q$, we have $L_b(c) \neq L_b(b)$. Apply Lemma 2.2 to define $r = f(b, c)$. Vertex r sees b, misses c, and $q < r$. Since b sees c, vertex r sees q by Lemma 2.4. Since r sees b, we have $r \neq d$. Since $f(e, b)$ is the neighbor of e on $\mathcal{R}$, it follows that $f(e, b) \leq q < r$, and r sees b so r sees e by Lemma 2.2. If r sees a, then there is a bull e-rab-c, a contradiction, so r misses a. If r misses d, then r, b, c, d, e is a hole, so r sees d. Suppose $\mathcal{R}$ has length > 3, then $f(e, b) < q = f(a, e) < r$, r sees e and misses a, a contradiction. So $\mathcal{R}$ has length 3, q sees e, and $q = f(e, b)$.

Since r sees e, misses a, and $e < a < r$, we have $L_a(e) \neq L_a(a)$. Apply Lemma 2.2

to define $s = f(a, e)$. Vertex s sees a, misses e, and $r < s$. Since s sees a, we have $s \neq d$. Since s misses e and $q = f(e, b) < r = f(b, c) < s$, it follows that s misses b, c by Lemma 2.2. If s sees d, then s, a, b, c, d is a hole, so s misses d. If s sees q, then b-asq-e is a bull, so s misses q. If s sees r, then c-der-s is a bull, so s misses r.

Since s sees a, misses q, and $a < q < s$, we have $L_q(a) \neq L_q(q)$. Apply Lemma 2.2 to define $t = f(q, a)$. Vertex t sees q, misses a, and $s < t$. Since q sees a, vertex t sees s by Lemma 2.4. Since t misses a and $q = f(e, b) < r = f(b, c) < s = f(a, e) < t$, vertex t misses b, c, e by Lemma 2.2. Since t misses c, we have $t \neq d$. If t sees r, then t, r, b, a, s is a hole, so t misses r, but then b-req-t is a bull, a contradiction. Thus the claim holds. ∎

Now we go on with the proof of the theorem. Since b sees c, misses e, and $c < e < b$, we have $L_e(c) \neq L_e(e)$. Apply Lemma 2.2 to define $p = f(e, c)$. Vertex p sees e, misses c, and $b < p$. Since p sees e and misses c, we have $p \neq a$ and $p \neq d$. If p sees a, then p sees the extremities of the P_5 a, b, c, d, e without seeing c, a contradiction to Lemma 3.1, so p misses a. If p sees b, then p sees d, otherwise p, b, c, d, e is a hole. If p misses b, then p misses d, otherwise the bad P_5 a-b-c-d-p is worse than a-b-c-d-e. So p either sees both b, d or misses both b, d.

CLAIM 2. $a < b$.

Proof. Suppose the claim is false, so $c < e < b < a$ by Claim 1.

Case 1. *$p < a$ and p sees b, d.* Since a sees b, misses p, and $b < p < a$, we have $L_p(b) \neq L_p(p)$. Apply Lemma 2.2 to define $q = f(p, b)$. Vertex q sees p, misses b, and $a < q$. Since p sees b, vertex q sees a by Lemma 2.4. Since q sees a, we have $q \neq d$. Since $p = f(e, c) < q$, vertex q either sees both e, c or misses both e, c. Suppose q misses e, c. If q sees d, then q, a, b, c, d is a hole, so q misses d. Then c-dep-q is a bull, a contradiction. So q sees e, c. Since q sees c, misses b, and $c < b < q$, we have $L_b(c) \neq L_b(b)$. Apply Lemma 2.2 to define $r = f(b, c)$. Vertex r sees b, misses c, and $q < r$. Since b sees c, vertex r sees q by Lemma 2.4. Since r sees b, misses c, and $p = f(e, c) < q = f(p, b) < r$, it follows that r sees p and misses e. But then c-brp-e is a bull, a contradiction.

Case 2. *$p < a$ and p misses b, d.* Since a sees b, misses p, and $b < p < a$, we can consider the chordless path $R = \mathcal{P}(a, p, b)$ of Lemma 2.3. If none of c, d, e has a neighbor in R^*, then the $R \cup \{c, d, e\}$ is a cycle of length at least 7, so one of c, d, e has a neighbor in R^*. Let q be the vertex of R^* closest to a that sees one of c, d, e. If q misses c, then one of $R[b, q] \cup \{c, d\}$, $R[b, q] \cup \{c, d, e\}$ is a hole of length ≥ 5. So q sees c. Since q sees c and $p = f(e, c) < q$, vertex q sees e. The hole $R[b, q] \cup \{c\}$ must have length < 5, so q sees a and so $q \neq d$. Since q sees c, misses b, and $c < b < q$, we have $L_b(c) \neq L_b(b)$. Apply Lemma 2.2 to define $r = f(b, c)$. Vertex r sees b, misses c, and $q < r$. Since b sees c, vertex r sees q by Lemma 2.4. Since r sees b, misses c, and $p = f(e, c) < f(p, b) \leq q < r$, vertex r sees p and misses e. Suppose $\mathcal{R}$ has length > 3, then $f(p, b) < q = f(a, p) < r$, r sees p, so r sees a and then c-bar-p is a bull, a contradiction. So $\mathcal{R}$ has length three and q sees p. But then q sees the extremities of the P_6 a-b-c-d-e-p without seeing b, a contradiction to Lemma 3.1.

Case 3. *$a < p$ and p sees b, d.* Since p sees b, misses a, and $b < a < p$, we have $L_a(b) \neq L_a(a)$. Apply Lemma 2.2 to define $q = f(a, b)$. Vertex q sees a, misses b, and $p < q$. Since a sees b, vertex q sees p by Lemma 2.4. Since q sees a, we have $q \neq d$. Since $p = f(e, c) < q$, vertex q either sees both e, c or misses both e, c. Suppose q misses e, c. If q sees d, then q, a, b, c, d is a hole, so q misses d. Then c-dep-q is a bull, a contradiction, so q sees c, e. Since q sees c, misses b, and $c < b < q$, we have $L_b(c) \neq L_b(b)$. Apply Lemma 2.2 to define $r = f(b, c)$. Vertex r sees b, misses c, and

$q < r$. Since b sees c, vertex r sees q by Lemma 2.4. Since r sees b, we have $r \neq d$. Since r sees b, misses c, and $p = f(e,c) < q = f(a,b) < r$, vertex r sees a and misses e. If r sees p, then c-brp-e is a bull, so r misses p. Since r sees a, misses p, and $a < p < r$, we have $L_p(a) \neq L_p(p)$. Apply Lemma 2.2 to define $s = f(p,a)$. Vertex s sees p, misses a and $r < s$. Since s misses a and $p = f(e,c) < q = f(a,b) < r = f(b,c) < s$, vertex s misses a,b,c,e. Since s misses c, we have $s \neq d$. If s misses d, then c-dep-s is a bull, so s sees d. But then a-b-c-d-s is a bad P_5 that is worse than a-b-c-d-e, a contradiction.

Case 4. *$a < p$ and p misses b,d.* Since p sees e, misses b, and $e < b < p$, we have $L_b(e) \neq L_b(b)$. Apply Lemma 2.2 to define $q = f(b,e)$. Vertex q sees b, misses e, and $p < q$. Since q sees b, we have $q \neq d$. Since q misses e and $p = f(e,c) < q$, vertex q misses c. If q misses d, then q-b-c-d-e is worse than a-b-c-d-e, so q sees d.

Case 4.1. *q misses a.* Since q sees b, misses a, and $b < a < q$, we have $L_a(b) \neq L_a(a)$. Apply Lemma 2.2 to define $r = f(a,b)$. Vertex r sees a, misses b, and $q < r$. Since a sees b, vertex r sees q by Lemma 2.4. Since r sees a, we have $r \neq d$. Since r misses b and $p = f(e,c) < q = f(b,e) < r$, vertex r misses b,c,e. If r sees d, then a,b,c,d,r is a hole, so r misses d. If r sees p, then a,b,c,d,e,p,r is a hole, so r misses p. Since r sees a, misses p, and $a < p < r$, we can consider the chordless path $R = \mathcal{P}(r,p,a)$ of Lemma 2.3. Every vertex u of R^* misses a and satisfies $p = f(e,c) < q = f(b,e) < r = f(a,b) < u$, so u misses a,b,c,e. If d has no neighbor in R^*, then $R \cup \{b,c,d,e\}$ is a cycle of length at least eight, so d has a neighbor in R^*. Let s be the vertex of R^* closest to a that sees d. Then $R[a,s] \cup \{b,c,d\}$ is a hole of length ≥ 5, a contradiction.

Case 4.2. *q sees a.* If q sees p, then c-baq-p is a bull, so q misses p. Since q sees a, misses p, and $a < p < q$, we can consider the chordless path $R = \mathcal{P}(q,p,a)$ of Lemma 2.3. Since $p = f(e,c) < q = f(b,e)$, every vertex of R^* either sees b,c,e or misses b,c,e. Let r be the neighbor of q in R^*. Vertex r misses a, and $f(p,a) \leq r$. If r misses b,c,e, then c-baq-r is a bull, so r sees b,c,e. Then a-bcr-e is a bull, a contradiction. Thus the claim holds. □

Claims 1 and 2 imply that $c < e < a < b$.

Since p sees e, misses a, and $e < a < p$, we have $L_a(e) \neq L_a(a)$. Apply Lemma 2.2 to define $q = f(a,e)$. Vertex q sees a, misses e, and $p < q$. Since q sees a, we have $q \neq d$. Since d sees e and misses a, it follows that $d < q = f(a,e)$. Since q misses e and $p = f(e,c) < q$, vertex q misses c. If q sees d, then q sees b, otherwise q,a,b,c,d is a hole. If q misses d, then q misses b, otherwise the bad P_5 q-b-c-d-e is worse than a-b-c-d-e. So q sees b,d or misses b,d.

CLAIM 3. *The path q-a-b-c-d-e-p is chordless.*

Proof. Suppose that q sees p. Then q sees the extremities of the path a-b-c-d-e-p without seeing c, so, by Lemma 3.1, the path is not chordless, so p sees b,d. If q misses b,d, then p sees the extremities of the path q-a-b-c-d-e without seeing c, a contradiction to Lemma 3.1, so q sees b,d. But then c-bqp-e is a bull, a contradiction. So q misses p.

Since q sees a, misses p, and $a < p < q$, we have $L_p(a) \neq L_p(p)$. Apply Lemma 2.2 to define $r = f(p,a)$. Vertex r sees p, misses a, and $q < r$. Since r misses a and $p = f(e,c) < q = f(a,e) < r$, vertex r misses c,e.

Suppose p sees b,d. If r misses d, then c-dep-r is a bull, so r sees d. If r misses b, then the bad P_5 a-b-c-d-r is worse than a-b-c-d-e, so r sees b. Then a-brp-e is a bull, a contradiction. So p misses b,d.

Suppose q sees b,d and r sees q. If r misses b, then c-baq-r is a bull, so r sees b.

If r misses d, then the bad P_5 r-b-c-d-e is worse than a-b-c-d-e, so r sees d. Then e-drq-a is a bull, a contradiction.

Suppose q sees b, d and r misses q. Since r sees p, misses q, and $p < q < r$, we have $L_q(p) \neq L_q(q)$. Apply Lemma 2.2 to define $s = f(q, p)$. Vertex s sees q, misses p, and $r < s$. Since s misses p and $p = f(e, c) < q = f(a, e) < r = f(p, a)$, vertex s misses a, c, e. If s misses b, then c-baq-s is a bull, so s sees b. If s misses d, then the bad P_5 s-b-c-d-e is worse than a-b-c-d-e, so s sees d. Then e-dsq-a is a bull, a contradiction. So q misses b, d. Thus the claim holds. □

Claim 4. $d < b$.

Proof. Suppose the claim is false, then $c < e < a < b < d$ by Claims 1 and 2.

Case 1. $L_d(b) \neq L_d(d)$. Apply Lemma 2.2 to define $s = f(d, b)$. Vertex s sees d, misses b, and $d < s$. Since s sees d, we have $s \neq p$. Suppose s sees c. If s misses e, then b-csd-e is a bull, so s sees e. If s misses a, then the bad P_5 a-b-c-s-e is worse than a-b-c-d-e, so s sees a. If s misses p, then a-sde-p is a bull, so s sees p. Then b-cds-p is a bull, a contradiction, so s misses c. If s sees a, then a, b, c, d, s is a hole, so s misses a. Then the bad P_5 a-b-c-d-s is worse than a-b-c-d-e, a contradiction.

Case 2. $L_d(b) = L_d(d)$. Since a sees b and misses d, we have $N(b) \neq N(d)$. Apply Lemma 2.5 to define $s = g(d, b)$. Vertex s sees d, misses b, $s < b$, and $L_d(s) < L_d(d)$. Since s misses b, we have $s \neq a, c$. Since q sees a and misses d, we have $L_d(a) \neq L_d(d)$, and since a sees b and misses d, we have $L'_d(a) \leq L'_d(s)$. If s sees q, then s sees the extremities of the P_5 q, a, b, c, d without seeing b, a contradiction to Lemma 3.1, so s misses q. So $L'_d(a) \neq L'_d(s)$. Apply Lemma 2.5 to define $t = h(s, a)$. Vertex t sees s, misses a, b, d, and $q < t$. Since t misses a and $p = f(e, c) < q = f(a, e) < t$, vertex t misses c, e. Since t misses e, we have $s \neq e$. If s sees c, then b-cds-t is a bull, so s misses c. If s sees a, then a, b, c, d, s is a hole, so s misses a. Suppose $s < e$. Since t sees s, misses e, and $s < e < t$, we have $L_e(s) \neq L_e(e)$. Apply Lemma 2.2 to define $u = f(e, s)$. Vertex u sees e, misses s, and $t < u$. Since u sees e and $p = f(e, c) < q = f(a, e) < t < u$, vertex u sees a, c. Since u sees a, misses s, $t = h(s, a) < u$, and $s = g(b, d)$, vertex u sees b, d. But then a-ucd-s is a bull, a contradiction. So $e < s$. Then the bad P_5 a-b-c-d-s is worse than a-b-c-d-e, a contradiction. Thus the claim holds. □

Claim 5. $L_b(d) = L_b(b)$.

Proof. Suppose the claim is false, so $L_b(d) \neq L_b(b)$. Apply Lemma 2.2 to define $s = f(b, d)$. Vertex s sees b, misses d, and $b < s$. Since s sees b, we have $s \neq q$. Suppose s sees c. If s misses a, then d-csb-a is a bull, so s sees a. If s misses e, then the bad P_5 a-s-c-d-e is worse than a-b-c-d-e, so s sees e. If s misses q, then e-sba-q is a bull, so s sees q. Then d-cbs-q is a bull, a contradiction, so s misses c. If s sees e, then b, c, d, e, s is a hole, so s misses e. Then s-b-c-d-e is a bad P_5 that is worse than a-b-c-d-e, a contradiction. Thus the claim holds. □

Claim 6. $a < d$.

Proof. Suppose the claim is false, then $d < a < b$. By Lemma 2.1, $L_b(d) \leq L_b(a) \leq L_b(b)$, and, by Claim 5, $L_b(d) = L_b(b)$, so $L_b(a) = L_b(b)$. Vertex q sees a, misses b, and $a < b < q$, a contradiction. Thus the claim holds. □

With the preceding claims, we have established that $c < e < a < d < b < p = f(e, c) < q = f(a, e)$, $L_b(d) = L_b(b)$, and q-a-b-c-d-e-p is a chordless path. Define sequences $(a_i), (b_i), (d_i), (e_i)$ as follows:

- $a_0 = a$, $b_0 = b$, $d_0 = d$, $e_0 = e$, $b_1 = q = f(a, e)$, $d_1 = p = f(e, c)$.
- For $i \geq 1$, $a_i = g(b_i, d_i)$, $e_i = g(d_i, b_{i-1})$.
- For $i \geq 2$, $b_i = h(a_{i-1}, e_{i-1})$, $d_i = h(e_{i-1}, a_{i-2})$.

For any $k \geq 1$, let us say that a-b-c-d-e admits an extension of order k, noted $\mathcal{W}_k$, if the sequences $(a_i)_{i<k}$, $(b_i)_{i\leq k}$, $(d_i)_{i\leq k}$, $(e_i)_{i<k}$ are well defined, and have the following property:

- $c < e_0 < a_0 < \cdots < e_{k-1} < a_{k-1} < d_0 < b_0 < \cdots < d_k < b_k$.
- $L_{b_{k-1}}(b_0) = \cdots = L_{b_{k-1}}(b_{k-1}) = L_{b_{k-1}}(d_0) = \cdots = L_{b_{k-1}}(d_{k-1})$.
- b_k-a_{k-1}-b_{k-1}-$\cdots$-b_1-a_0-b_0-c-d_0-e_0-d_1-$\cdots$-d_{k-1}-e_{k-1}-d_k is a chordless path.

Claims 1–6 and the definition of p, q shows that a-b-c-d-e admits an extension of order 1. Let k be the greatest integer such that a-b-c-d-e admits an extension $\mathcal{W}_k$ of order k. We will prove that a-b-c-d-e admits an extension of order $k+1$. Since G is finite, this is a contradiction that will complete the proof that there is no bad P_5.

CLAIM 7. $L_{d_k}(b_{k-1}) = L_{d_k}(d_k)$.

Proof. For suppose that $L_{d_k}(b_{k-1}) \neq L_{d_k}(d_k)$. Since $b_{k-1} < d_k$ we can apply Lemma 2.2 to define $r = f(d_k, b_{k-1})$. Vertex r sees d_k, misses b_{k-1}, and $d_k < r$. Since r sees d_k, we have $r \neq b_k$. Since r misses b_{k-1} and $L_{b_{k-1}}(b_0) = \cdots = L_{b_{k-1}}(b_{k-1}) = L_{b_{k-1}}(d_0) = \cdots = L_{b_{k-1}}(d_{k-1})$, it follows that r misses $b_0, \ldots, b_{k-1}, d_0, \ldots, d_{k-1}$. Since r misses $b_0, \ldots, b_{k-1}, d_0, \ldots, d_{k-1}$ and $e_1 = g(d_1, b_0) < a_1 = g(b_1, d_1) < \cdots < a_{k-2} = g(b_{k-2}, d_{k-2}) < e_{k-1} = g(d_{k-1}, b_{k-2}) < d_1 = f(e_0, c) < b_1 = f(a_0, e_0) < d_2 = h(e_1, a_0) < b_2 = h(a_1, e_1) < \cdots < b_{k-1} = h(a_{k-2}, e_{k-2}) < d_k = h(e_{k-1}, a_{k-2}) < r$, it follows that r either sees all of $c, a_0, \ldots, a_{k-2}, e_0, \ldots, e_{k-1}$ or misses all of them. If r sees them, then d_{k-1}-$e_{k-1}d_k r$-a_{k-2} is a bull, so r misses them. If r sees one of a_{k-1}, b_k, then $\mathcal{W}_k \cup \{r\}$ contains a hole of length at least six, a contradiction, so r misses a_{k-1}, b_k.

Case 1. $r < b_k$. Since b_k sees a_{k-1}, misses b_{k-1}, and $a_{k-1} < b_{k-1} < b_k$, we have $L_{b_{k-1}}(a_{k-1}) \neq L_{b_{k-1}}(b_{k-1})$. Apply Lemma 2.2 to define $s = f(b_{k-1}, a_{k-1})$. Vertex s sees b_{k-1}, misses a_{k-1} and $b_k < s$. Since b_{k-1} sees a_{k-1}, vertex s sees b_k by Lemma 2.4. Since s sees b_{k-1} and $r = f(d_k, b_{k-1}) < b_k < s$, vertex s sees d_k. Since s sees b_k, d_k and misses a_{k-1}, it follows from Lemma 3.1 that r sees all of $b_0, \ldots, b_k, d_0, \ldots, d_k$ and misses all of c, $a_0, \ldots, a_{k-1}, e_0, \ldots, e_{k-1}$. If s sees r, then e_{k-1}-$d_k rs$-b_k is a bull, so s misses r.

Since s sees d_k, misses r, and $d_k < r < s$, we have $L_r(d_k) \neq L_r(r)$. Apply Lemma 2.2 to define $t = f(r, d_k)$. Vertex t sees r, misses d_k, and $s < t$. Since r sees d_k, vertex t sees s by Lemma 2.4. Since t misses d_k and $r = f(d_k, b_{k-1}) < s = f(b_{k-1}, a_{k-1}) < t$, vertex t misses a_{k-1}, b_{k-1}. Since t misses b_{k-1} and $L_{b_{k-1}}(b_0) = \cdots = L_{b_{k-1}}(b_{k-1}) = L_{b_{k-1}}(d_0) = \cdots = L_{b_{k-1}}(d_{k-1})$, it follows that t misses all of $b_0, \ldots, b_{k-1}, d_0, \ldots, d_{k-1}$. Since t misses $a_{k-1}, b_0, \ldots, b_{k-1}, d_0, \ldots, d_k$, and $e_1 = g(d_1, b_0) < a_1 = g(b_1, d_1) < \cdots < e_{k-1} = g(d_{k-1}, b_{k-2}) < a_{k-1} = g(b_{k-1}, d_{k-1}) < d_1 = f(e_0, c) < b_1 = f(a_0, e_0) < d_2 = h(e_1, a_0) < b_2 = h(a_1, e_1) < \cdots < d_k = h(e_{k-1}, a_{k-2}) < b_k = h(a_{k-1}, e_{k-1}) < t$, it follows that t misses all of $c, a_0, \ldots, a_{k-1}, e_0, \ldots, e_{k-1}$.

Since t sees r, misses b_k, and $r < b_k < t$, we can consider the chordless path $R = \mathcal{P}(t, b_k, r)$ of Lemma 2.3. Every vertex u of R^* misses r and satisfies $t = f(r, d_k) < u$, so u misses d_k. The cycle $R \cup \mathcal{W}_k$ has length at least ten, so one of $\mathcal{W}_k \setminus \{b_k\}$ has a neighbor in R^*. Let u be the vertex of R^* closest to t that sees one of $\mathcal{W}_k \setminus \{b_k\}$, then $R[u, r] \cup \mathcal{W}_k$ contains a hole of size ≥ 5, a contradiction.

Case 2. $b_k < r$. Since r sees d_k, misses b_k, and $d_k < b_k < r$, we can consider the chordless path $R = \mathcal{P}(r, b_k, d_k)$ of Lemma 2.3. Every vertex u of R^* misses d_k and satisfies $r = f(d_k, b_{k-1}) < u$, so u misses b_{k-1}. Then, since $L_{b_{k-1}}(b_0) = \cdots = L_{b_{k-1}}(b_{k-1}) = L_{b_{k-1}}(d_0) = \cdots = L_{b_{k-1}}(d_{k-1})$, vertex u misses all of $b_0, \ldots, b_{k-1}$, $d_0, \ldots, d_{k-1}$. Since u misses $b_0, \ldots, b_{k-1}, d_0, \ldots, d_k$ and $e_1 = g(d_1, b_0) < a_1 =$

$g(b_1,d_1) < \cdots < e_{k-1} = g(d_{k-1},b_{k-2}) < a_{k-1} = g(b_{k-1},d_{k-1}) < d_1 = f(e_0,c) < b_1 = f(a_0,e_0) < d_2 = h(e_1,a_0) < b_2 = h(a_1,e_1) < \cdots d_k = h(e_{k-1},a_{k-2}) < b_k = h(a_{k-1},e_{k-1}) < u$, vertex u either sees all of $c, a_0, \dots, a_{k-1}, e_0, \dots, e_{k-1}$ or misses all of them.

Let t be the neighbor of b_k in R^*, so $t = f(b_k, d_k)$. If t sees $c, a_0, \dots, a_{k-1}, e_0, \dots, e_{k-1}$, then b_{k-1}-$a_{k-1}b_kt$-e_{k-1} is a bull. So t misses $c, a_0, \dots, a_{k-1}$, $e_0, \dots, e_{k-1}$. If t sees r, then $\mathcal{W}_k \cup \{r,t\}$ is a hole, so t misses r.

Let u be the neighbor of r in R^*, so $u = f(r, b_k)$. Vertex u misses $b_0, \dots, b_k, d_0, \dots, d_k$. If u misses $c, a_0, \dots, a_{k-1}, e_0, \dots, e_{k-1}$, then $R \cup \mathcal{W}_k$ contains a hole of size ≥ 5, so u sees $c, a_0, \dots, a_{k-1}, e_0, \dots, e_{k-1}$.

Since u sees c, misses b_0, and $c < b_0 < u$, we have $L_{b_0}(c) \neq L_{b_0}(b_0)$. Apply Lemma 2.2 to define $v = f(b_0, c)$. Vertex v sees b_0, misses c, and $u < v$. Since b_0 sees c, vertex v sees u by Lemma 2.4. Since v sees b_0 and $L_{b_{k-1}}(b_0) = \cdots = L_{b_{k-1}}(b_{k-1}) = L_{b_{k-1}}(d_0) = \cdots = L_{b_{k-1}}(d_{k-1})$, vertex v misses $b_0, \dots, b_{k-1}, d_0, \dots, d_{k-1}$. Since v sees b_{k-1}, misses c, and $d_1 = f(e_0, c) < r = f(d_k, b_{k-1}) < t = f(b_k, d_k) < u = f(r, b_k) < v$, vertex v sees d_k, b_k, r and misses e_0. But then b_0-vru-e_0 is a bull, a contradiction. Thus the claim holds. □

CLAIM 8. $L_{d_k}(b_0) = \cdots = L_{d_k}(b_{k-1}) = L_{d_k}(d_0) = \cdots = L_{d_k}(d_k)$.

Proof. By Claim 7, $L_{d_k}(b_{k-1}) = L_{d_k}(d_k)$, and $L_{b_{k-1}}(b_0) = \cdots = L_{b_{k-1}}(b_{k-1}) = L_{b_{k-1}}(d_0) = \cdots = L_{b_{k-1}}(d_{k-1})$, and $b_{k-1} < d_k$, so $L_{d_k}(b_0) = \cdots = L_{d_k}(b_{k-1}) = L_{d_k}(d_0) = \cdots = L_{d_k}(d_k)$. Thus the claim holds. □

Since a_{k-1} sees b_{k-1} and misses d_k, we have $N(b_{k-1}) \neq N(d_k)$. Apply Lemma 2.5 to define $e_k = g(d_k, b_{k-1})$. Vertex e_k sees d_k, misses b_{k-1}, $e_k < b_{k-1}$, and $L_{d_k}(e_k) < L_{d_k}(d_k) = L_{d_k}(d_0)$. Since $L_{d_k}(e_k) < L_{d_k}(d_0)$, so $e_k < d_0$ by Lemma 2.1. Since e_k sees d_k, so $e_k \notin \mathcal{W}_k \setminus \{e_{k-1}\}$. Since a_{k-1} sees b_{k-1} and misses d_k, we have $L'_{d_k}(a_{k-1}) \leq L'_{d_k}(e_k)$. If e_k sees b_k, then e_k sees the extremities of the chordless path $\mathcal{W}_k$ without seeing b_{k-1}, a contradiction to Lemma 3.1, so e_k misses b_k. So $L'_{d_k}(a_{k-1}) < L'_{d_k}(e_k)$. Apply Lemma 2.5 to define $d_{k+1} = h(e_k, a_{k-1})$. Vertex d_{k+1} sees e_k, misses a_{k-1}, b_{k-1}, d_k, and $b_k < d_{k+1}$.

CLAIM 9. *$\mathcal{W}_k$-e_k-d_{k+1} is a chordless path.*

Proof. Since d_{k+1} misses d_k and $L_{d_k}(b_0) = \cdots = L_{d_k}(b_{k-1}) = L_{d_k}(d_0) = \cdots = L_{d_k}(d_k)$, vertex d_{k+1} misses $b_0, \dots, b_{k-1}, d_0, \dots, d_k$. Since d_{k+1} misses a_{k-1}, $b_0, \dots,$ b_{k-1}, $d_0, \dots, d_k$, and $e_1 = g(d_1, b_0) < a_1 = g(b_1, d_1) < \cdots < e_{k-1} = g(d_{k-1}, b_{k-2}) < a_{k-1} = g(b_{k-1}, d_{k-1}) < d_1 = f(e_0, c) < b_1 = f(a_0, e_0) < d_2 = h(e_1, a_0) < b_2 = h(a_1, e_1) < \cdots d_k = h(e_{k-1}, a_{k-2}) < b_k = h(a_{k-1}, e_{k-1}) < t$, vertex d_{k+1} misses $c, a_0, \dots, a_{k-1}, e_0, \dots, e_{k-1}$. Since d_{k+1} misses e_{k-1}, we have $e_k \neq e_{k-1}$. If d_{k+1} sees b_k, then e_k sees the extremities of the chordless path $\mathcal{W}_k \cup \{d_{k+1}\}$ without seeing b_{k-1}, a contradiction to Lemma 3.1, so d_{k+1} misses b_k.

Suppose e_k sees d_{k-1}. Consider the general step of the algorithm when b_{k-1} is chosen. Since $L_{d_k}(e_k) < L_{d_k}(d_k) = L_{d_k}(b_{k-1})$, we have $L_{b_{k-1}}(e_k) < L_{b_{k-1}}(b_{k-1})$, by Lemma 2.1. Since $L'_{d_k}(a_{k-1}) < L'_{d_k}(e_k)$, and $L_{d_k}(b_{k-1}) = L_{d_k}(d_k)$, we have $L'_{b_{k-1}}(a_{k-1}) < L'_{b_{k-1}}(e_k)$. Set U of step 1 of the algorithm contains e_k because $L_{b_{k-1}}(e_k) < L_{b_{k-1}}(b_{k-1})$. Since $L'_{b_{k-1}}(a_{k-1}) < L'_{b_{k-1}}(e_k)$, vertex e_k is selected from U at step 2.1 before a_{k-1}. Then at step 2.2, $A \cap N(e_k)$ must be empty, for otherwise b_{k-1} is removed from A and b_{k-1} is not the selected vertex at step 3. Since vertex d_{k-1} is in $N(e_k)$, it has been removed earlier from A by a vertex u with $L'_{b_{k-1}}(e_k) \leq L'_{b_{k-1}}(u)$. Since $L'_{b_{k-1}}(u) \geq L'_{b_{k-1}}(e_k) > L'_{b_{k-1}}(a_{k-1})$, we have $u \neq a_{k-1}$. This contradicts the definition of a_{k-1}, so e_k misses d_{k-1}.

If e_k sees e_{k-1}, then d_{k-1}-$e_{k-1}d_ke_k$-d_{k+1} is a bull, so e_k misses e_{k-1}. If e_k sees one of $b_0, \dots, b_{k-2}, d_0, \dots, d_{k-2}, c, a_0, \dots, a_{k-1}, e_0, \dots, e_{k-2}$, then $\mathcal{W}_k \cup \{s\}$ contains a hole of length > 5, so e_k missees $b_0, \dots, b_{k-2}, d_0, \dots, d_{k-2}, c, a_0, \dots, a_{k-2}, e_0, \dots, e_{k-2}$. Thus the claim holds. ◻

CLAIM 10. $a_{k-1} < e_k$.

Proof. Suppose the claim is false and $e_k < a_{k-1}$. Since d_{k+1} sees e_k, misses a_{k-1}, and $e_k < a_{k-1} < d_{k+1}$, we have $L_{a_{k-1}}(e_k) \neq L_{a_{k-1}}(a_{k-1})$. Apply Lemma 2.2 to define $u = f(a_{k-1}, e_k)$. Vertex u sees a_{k-1}, misses e_k, and $d_{k+1} < u$. Since u sees a_{k-1}, misses e_k, $d_{k+1} = h(e_k, a_{k-1}) < u$, and $e_k = g(d_k, b_{k-1})$, vertex u sees d_k, b_{k-1}. Since u sees the extremities of the chordless path $\mathcal{W}_k \setminus \{b_k\}$, by Lemma 3.1 it must see all the vertices of $\mathcal{W}_k \setminus \{b_k\}$. But then a_{k-1}-$ue_{k-1}d_k$-e_k is a bull, a contradiction. Thus the claim holds. ◻

Claims 8, 9, and 10, and the definition of e_k, d_{k+1}, show that the sequences $(a_i)_{i<k}$, $(b_i)_{i\leq k}$, $(d_i)_{i\leq k+1}$, $(e_i)_{i<k+1}$ are well defined and satisfy the following properties:

- $c < e_0 < a_0 < \cdots < e_{k-1} < a_{k-1} < e_k < d_0 < b_0 < \cdots < d_k < b_k < d_{k+1}$.
- $L_{d_k}(b_0) = \cdots = L_{d_k}(b_{k-1}) = L_{d_k}(d_0) = \cdots = L_{d_k}(d_k)$.
- $\mathcal{W}_k$-e_k-d_{k+1} is a chordless path.

The same type of proof can be done (and we omit the details) to define vertices $a_k = g(b_k, d_k)$ and $b_{k+1} = h(a_k, e_k)$ and to show that they satisfy the following properties:

- $c < e_0 < a_0 < \cdots < e_k < a_k < d_0 < b_0 < \cdots < d_{k+1} < b_{k+1}$.
- $L_{b_k}(b_0) = \cdots = L_{b_k}(b_k) = L_{b_k}(d_0) = \cdots = L_{b_k}(d_k)$.
- b_{k+1}-a_k-$\mathcal{W}_k$-e_k-d_{k+1} is a chordless path.

This means that a-b-c-d-e admits an extension of order $k+1$. This is a contradiction to the definition of k. This completes the proof of the theorem. ◻

4. Algorithm COSINE*.

Algorithm COSINE* is a particular case of Algorithm COSINE due to Hertz [20], which is an $\mathcal{O}(nm)$ algorithm for optimally coloring the vertices of a Meyniel graph. The difference between COSINE and COSINE* is that the input graph of COSINE* has an ordering σ on its vertices and ties are broken according to this ordering.

Colors are viewed as integers $1, 2, \dots, \ell$. Algorithm COSINE* constructs the color classes iteratively. To construct the class of color c, the algorithm selects vertices until all the vertices of the graph have a neighbor colored c. At each step, the vertex that is selected and colored c is the vertex that has no neighbor already colored c and has the maximum number of uncolored neighbors in common with the vertices already colored c, with ties being broken by taking such a vertex that minimizes σ. More formally:

ALGORITHM COSINE*
Input: A graph G on n vertices and an ordering σ on its vertices.
Output: A coloring of the vertices of G.
Initialization: $c = 1$;
General step: While there exist uncolored vertices do:
1. While there exist uncolored vertices that have no neighbor colored c do:
   1.1. Let A be the set of uncolored vertices that have a neighbor colored c;
   1.2. Select an uncolored vertex u that has no neighbor colored c and has the maximum number of neighbors in A, with ties being broken by taking such a vertex that is minimum for σ;
   1.3. Color u with c;

2. $c := c + 1$.

One may remark that the original formulation of Algorithm Cosine in [20] is different. Hertz explains his algorithm in terms of vertex contraction. We prefer to modify the formulation of the algorithm to simplify the algorithmic concepts. To prove the optimality of the algorithm, we need to introduce the notion of contraction, which is done in the next section.

Complexity analysis. To analyze the complexity of algorithm Cosine*, we will assume that the input graph is connected; thus if n is the number of vertices and m the number of edges of the graph, we have $m \geq n-1$. If the graph is not connected, then it suffices to apply the algorithm on each of its components. Breaking the ties in Cosine* does not increase the complexity of Algorithm Cosine, that is, it can be implemented in time $\mathcal{O}(nm)$ as follows. Updating the set A at step 1.1 can be done in time $\mathcal{O}(d(u))$ whenever a new vertex u is colored at step 1.3, by adding the uncolored neighbors of u to A. For one given color c, this procedure takes time $\mathcal{O}(n+m)$, so the total cost is $\mathcal{O}(nm)$ over all colors. To compute step 1.2 efficiently, we use for each vertex a counter that represents the number of its neighbors in A. Every time a vertex is added to A we update the counter of the other vertices; this can also be done in time $\mathcal{O}(n+m)$ for a given color and so in time $\mathcal{O}(nm)$ over all colors. Then we search all the vertices in time $\mathcal{O}(n)$ to find the uncolored vertex that has the maximum counter and is minimum for σ. After each such search, one vertex is colored, so the total cost of all such searches is $\mathcal{O}(n^2)$. Therefore, the total running time of Algorithm Cosine* is $\mathcal{O}(nm)$.

5. Even pairs contraction. An *even pair* in a graph G is a pair of nonadjacent vertices such that every chordless path between them has even length. A survey on even pairs is given in [12]. Given two nonadjacent vertices x, y in G, the operation of *contracting* them means removing x and y and adding one vertex with an edge to each vertex of $N(x) \cup N(y)$. The following lemmas state essential results about even pairs.

Lemma 5.1 (see [13, 29]). *For any graph G, the graph G' obtained from G by contracting an even pair of G satisfies $\omega(G') = \omega(G)$ and $\chi(G') = \chi(G)$.*

Lemma 5.2 (see [12]). *If a graph G contains no odd hole, then the graph G' obtained from G by contracting an even pair contains no odd hole.*

Lemma 5.3 (see [12]). *If a graph G contains no antihole, then the graph G' obtained from G by contracting an even pair contains no antihole different from $\overline{C}_6$.*

Following Bertschi [4], a graph G is called *even contractile* if it is either a clique or it contains an even pair whose contraction yields an even contractile graph, and G is *perfectly contractile* if every induced subgraph of G is even contractile. See [12] for a survey on perfectly contractile graphs.

We need to define a superclass of $\mathcal{B}$. Let us say that a graph G is a *quasi-$\mathcal{B}$* graph if G is a Berge graph that contains no antihole of length at least five and G has a vertex, called a *pivot*, that is an ear of every bull of G. (This definition can be compared with the definition of quasi-Meyniel graphs in [20].) We observe that every graph in class $\mathcal{B}$ is a quasi-$\mathcal{B}$ graph (and in such a graph, every vertex is a pivot), and if G is a quasi-$\mathcal{B}$ graph and z is a pivot, then $G \setminus z$ is in class $\mathcal{B}$.

We prove that, for every graph G in class $\mathcal{B}$, Algorithm LexBFS* applied on $\overline{G}$ followed by Algorithm Cosine* applied on G produces a coloring of the vertices of G with $\omega(G)$ colors, where $\omega(G)$ is the maximum size of a clique in G. This will prove the optimality of this algorithm on the class $\mathcal{B}$. Our proof follows the same steps as Hertz's proof [20] that his algorithm Cosine is optimal on quasi-Meyniel graphs. Just like in [20], the optimality of our algorithm will follow from the fact that each color class produced by the algorithm corresponds to the contraction of even pairs.

The following lemma generalizes Lemma 3.1 to quasi-$\mathcal{B}$ graphs.

LEMMA 5.4. *In a* quasi-$\mathcal{B}$ *graph* G, *let* $P = a_0\text{-}a_1\text{-}\cdots\text{-}a_r$ *be a chordless odd path with* $r \geq 5$, *where* a_0 *is a pivot of* G, *and let* u *be a vertex that sees the two endvertices* a_0, a_r *of* P. *Then* u *sees* a_2.

Proof. Suppose the lemma is false and u misses a_2. If u sees a_1, then $a_r\text{-}ua_0a_1\text{-}a_2$ is a bull of which a_0 is not an ear, a contradiction. So u misses a_1. Denote a *segment* as any subpath of P, of length at least one, whose endvertices see u and interior vertices do not. So P is (edgewise) partitioned into its segments. Since G is odd-hole-free, every segment has length one or even length. Since P is odd, there is a least one segment of length one. Let i be the smallest integer such that u sees a_i and a_{i+1}. Since u misses a_1, a_2, we have $i \geq 3$. Then $a_{i-1}\text{-}a_ia_{i+1}u\text{-}a_0$ is a bull of which a_0 is not an ear, a contradiction. ∎

Now we prove the following theorem, which implies the optimality of our coloring algorithm.

THEOREM 5.5. *Let* G *be in class* $\mathcal{B}$. *Then the coloring obtained by Algorithm* LEXBFS* *applied on* $\overline{G}$ *followed by Algorithm* COSINE* *applied on* G *uses exactly* $\omega(G)$ *colors.*

Proof of Theorem 5.5. Let ℓ be the total number of colors used by the algorithm. For each color $c \in \{1, \dots, \ell\}$ let k_c be the number of vertices colored c. Therefore every vertex of G can be renamed x_c^i, where $c \in \{1, \dots, \ell\}$ is the color assigned to the vertex by the algorithm and $i \in \{1, \dots, k_c\}$ is the integer such that x_c^i is the ith vertex colored c. Thus $V(G) = \{x_1^1, x_1^2, \dots, x_1^{k_1}, x_2^1, \dots, x_2^{k_2}, \dots, x_\ell^1, \dots, x_\ell^{k_\ell}\}$.

Define a sequence of graphs and vertices as follows. Put $G_1^1 = G$ and $w_1^1 = x_1^1$ (that is a pivot of G). For $i = 2, \dots, k_1$, call G_1^i the graph obtained from G_1^{i-1} by contracting w_1^{i-1} and x_1^i into a new vertex w_1^i colored with the color one. In the graph $G_1^{k_1}$, we remark that $w_1^{k_1}$ is adjacent to all other vertices of $G_1^{k_1}$; for otherwise, there is a vertex y that is not adjacent to $w_1^{k_1}$, that means that y has no neighbor of color one, so the algorithm should have colored more vertices with color one; a contradiction. More simply, let us call w_1 the vertex $w_1^{k_1}$.

The sequence continues as follows. For each $c \in \{2, \dots, \ell\}$, put $G_c^1 = G_{c-1}^{k_{c-1}}$ and $w_c^1 = x_c^1$. For $i = 2, \dots, k_c$, call G_c^i the graph obtained from G_c^{i-1} by contracting vertices w_c^{i-1} and x_c^i into a new vertex w_c^i colored with the color c. In $G_c^{k_c}$, we can again remark that $w_c^{k_c}$ is adjacent to all other vertices of $G_c^{k_c}$, for the same reason as above, and we simply call w_c the vertex $w_c^{k_c}$. So the last graph in the sequence, $G_\ell^{k_\ell}$, is a clique of size l with vertices $w_1, \dots, w_\ell$, where each w_c is obtained by the contraction of the vertices of color c.

CLAIM 1. *For every color* $c \in \{1, \dots, \ell\}$ *and integer* $i \in \{1, \dots, k_c - 1\}$, *if* G_c^i *is a* quasi-$\mathcal{B}$ *graph,* w_c^i *is a pivot, and not the top of a house of* G_c^i, *then there is no chordless odd path from* w_c^i *to* x_c^{i+1} *in* G_c^i.

Proof. Suppose on the contrary that there exists a chordless odd path $P = a_0\text{-}a_1\text{-}\cdots\text{-}a_{r-1}\text{-}a_r$ from $a_0 = w_c^i$ to $a_r = x_c^{i+1}$ in G_c^i. We have $r \geq 3$ since w_c^i, x_c^{i+1} are not adjacent. Note that every vertex of P has a nonneighbor in G_c^i. Put $W_1 = \emptyset$ and $W_c = \{w_1, \dots, w_{c-1}\}$ if $c \geq 2$, and recall that any $w \in W_c$ is a vertex of G_c^i that is adjacent to all vertices of $G_c^i \setminus w$. So P contains no vertex of W_c. We know that every vertex of $G_c^i \setminus W_c$ will have a color from $\{c, c+1, \dots, \ell\}$ when the algorithm terminates.

Let us consider the situation when Algorithm COSINE* selects x_c^{i+1}. Let A be the set defined at step 1.1 of the algorithm. Vertex a_1 is in A and a_2 is not in A. Let $T = N(x_c^{i+1}) \cap A$. Every vertex of T is adjacent to at least one vertex colored c in G and thus is adjacent to w_c^i in G_c^i.

Suppose that there exists a vertex $t \in T$ that misses a_2. If $r = 3$, then either t misses a_1 and then u, a_0, a_1, a_2, a_3 induce an odd hole, or t sees a_1 and then a_0 is the top of a house, in either case a contradiction. So $r \geq 5$. Vertex t sees both extremities of the chordless odd path P without seeing a_2, a contradiction to Lemma 5.4. So every vertex of T sees a_2. Then $T \cup \{a_1\} \subset N(a_2) \cap A$, and so a_2 has strictly more neighbors in A than x_c^{i+1}, which contradicts the fact that x_c^{i+1} is selected at step 1.2. Thus the claims holds. ∎

CLAIM 2. *For every color $c \in \{1, \ldots, \ell\}$ and integer $i \in \{0, 1, \ldots, k_c - 1\}$, the following two properties hold*:

(A_i) *If $i \geq 1$, then w_c^i and x_c^{i+1} form an even pair of G_c^i.*

(B_i)
1. *G_c^{i+1} is a quasi-$\mathcal{B}$ graph.*
2. *w_c^{i+1} is a pivot of G_c^{i+1}.*
3. *w_c^{i+1} is not the top of a house of G_c^{i+1}.*

Proof. Let $c \in \{1, \ldots, \ell\}$. We show by induction on i that (A_i) and (B_i) hold.

Property (A_0) holds by vacuity. Graph G_1^1 is in $\mathcal{B}$, so w_c^1 is a pivot of this graph, and so (1) and (2) are satisfied when $c = 1$ and $i = 0$. To prove item 3, consider the beginning of Algorithm COSINE*: The set A of step 1.1 is empty, so w_1^1 is the minimum vertex of σ. Since the ordering σ was obtained by Algorithm LEXBFS* applied on $\overline{G}$, Theorem 3.2 ensures that w_1^1 is not the middle of a P_5 in $\overline{G_1^1}$, so w_1^1 is not the top of a house in G_1^1.

Suppose $c \geq 2$. In the graph G_c^1, every vertex w_h with $h \in \{1, \ldots, c-1\}$ is adjacent to all other vertices of the graph; moreover, $G_c^1 \setminus \{w_1, \ldots, w_{c-1}\}$ is in $\mathcal{B}$, since it is a subgraph of G. It follows that G_c^1 is actually in $\mathcal{B}$, and so w_c^1 is a pivot of this graph. At this step of Algorithm COSINE* the set A of step 1.1 is empty, so at step 1.2 every vertex of $G_c^1 \setminus \{w_1, \ldots, w_{c-1}\}$ has no neighbor colored c and has the maximum number of neighbors in A, so the vertex $w_c^1 = x_c^1$ that is selected is the minimum for σ in $G_c^1 \setminus \{w_1, \ldots, w_{c-1}\}$, and Theorem 3.2 ensures that this vertex is not the top of a house in $G_c^1 \setminus \{w_1, \ldots, w_{c-1}\}$. Since every vertex w_h with $h \in \{1, \ldots, c-1\}$ is adjacent to all other vertices of the graph, it follows that w_c^1 is not the top of a house in G_c^1.

Now suppose that $i \geq 1$ and that (A_{i-1}) and (B_{i-1}) hold. Claim 1 implies immediately that (A_i) holds. It remains to prove (B_i). By (A_i), (B_{i-1}), and Lemmas 5.2 and 5.3, the graph G_c^{i+1} contains no odd hole and no antihole different from $\overline{C}_6$.

Suppose that G_c^{i+1} contains a $\overline{C}_6$, with vertices $a_1, a_2, a_3, a_4, a_5, a_6$ and nonedges a_1a_2, a_2a_3, a_3a_4, a_4a_5, a_5a_6, a_6a_1. If w_c^{i+1} is not one of the a_i's, then this $\overline{C}_6$ is also contained in G_c^i, a contradiction. So, by symmetry, we may assume that $w_c^{i+1} = a_1$. By the definition of contraction, both w_c^i, x_c^{i+1} miss a_6 and a_2, and each of a_3, a_4, a_5 sees at least one of w_c^i, x_c^{i+1}. At least one of w_c^i, x_c^{i+1} sees both a_3, a_5, for otherwise either w_c^i-a_3-a_5-x_c^{i+1} or w_c^i-a_5-a_3-x_c^{i+1} is a chordless path between w_c^i and x_c^{i+1}, a contradiction to (A_i). Call u a vertex of w_c^i, x_c^{i+1} that sees both a_3, a_5, and call v the other one. None of u, v sees all of a_3, a_4, a_5, for otherwise a $\overline{C}_6$ is contained in G_c^i. So u misses a_4, and so v sees a_4 and misses at least one of a_3, a_5. By symmetry we can assume that v misses a_3. But then v-$a_4a_2a_6$-a_3 is a bull of G_c^i of which w_c^i is not an ear, a contradiction. So G_c^{i+1} contains no $\overline{C}_6$.

Suppose that G_c^{i+1} contains a bull a_1-$a_2a_3a_4$-a_5 such that w_c^{i+1} is not an ear of this bull. If w_c^{i+1} is not in the bull, then the bull is also contained in G_c^i and w_c^i is not in it, which contradicts the fact that w_c^i is a pivot of G_c^i. So, by symmetry, we may assume that $w_c^{i+1} = a_1$ or $w_c^{i+1} = a_3$. If $w_c^{i+1} = a_1$, then w_c^i, x_c^{i+1} miss all of a_3, a_4, a_5, and at least one of w_c^i, x_c^{i+1} sees a_2; but this yields a bull in G_c^i of which w_c^i is not an ear, a contradiction. If $w_c^{i+1} = a_3$, then both w_c^i, x_c^{i+1} miss both a_1, a_5, and at least one of w_c^i, x_c^{i+1} sees both a_2, a_4, for otherwise either w_c^i-a_2-a_4-x_c^{i+1} or w_c^i-a_4-a_2-x_c^{i+1} is a chordless path between w_c^i and x_c^{i+1}, a contradiction to (A_i). But this yields a bull in G_c^i of which w_c^i is not an ear, a contradiction.

It follows from the preceding two paragraphs that G_c^{i+1} is a quasi-$\mathcal{B}$ graph and that w_c^{i+1} is a pivot of G_c^{i+1}.

Now suppose that w_c^{i+1} is the top of a house in G_c^{i+1} with vertices a_1, a_2, a_3, a_4, a_5 and nonedges $a_1a_2, a_2a_3, a_3a_4, a_4a_5$. So $w_c^{i+1} = a_3$. In G_c^i, both w_c^i, x_c^{i+1} miss a_2, a_4. Vertex w_c^i misses at least one of a_1, a_5, for otherwise it is the top of a house in G_c^i, a contradiction to (B_{i-1}). By symmetry, we may assume that w_c^i misses a_5, and so x_c^{i+1} sees a_5. Then x_c^{i+1} also sees a_1, for otherwise w_c^i-a_1-a_5-x_c^{i+1} is a path that contradicts (A_i). Then w_c^i misses a_1, for otherwise w_c^i-$a_1x_c^{i+1}a_5$-a_2 is a bull in G_c^i of which w_c^i is not an ear. Note that, in G_c^i, vertices $a_1, a_2, x_c^{i+1}, a_4, a_5$ induce a house, of which x_c^{i+1} is the top, and w_c^i misses all of them. Let us consider the situation when Algorithm COSINE* selects x_c^{i+1}. Let A be the set defined at step 1.1 of the algorithm. Since w_c^i misses all of the a_i's, none of them is in A. Let $T = N(x_c^{i+1}) \cap A$, and consider any vertex t of T. By the definition of T, vertex t sees x_c^{i+1} and w_c^i in G_c^i. If t misses both a_1, a_5, then t sees a_4, for otherwise t-$x_c^{i+1}a_5a_1$-a_4 is a bull in G_c^i of which w_c^i is not an ear, and similarly t sees a_2, but then w_c^i-ta_4a_2-a_5 is a bull in G_c^i of which w_c^i is not an ear. So t sees at least one of a_1, a_5, say a_1. Then t sees a_4, for otherwise w_c^i-$tx_c^{i+1}a_1$-a_4 is a bull in G_c^i of which w_c^i is not an ear. Then t sees a_2, for otherwise w_c^i-ta_1a_4-a_2 is a bull in G_c^i of which w_c^i is not an ear. Then t sees a_5, for otherwise w_c^i-ta_4a_2-a_5 is a bull in G_c^i of which w_c^i is not an ear. So every vertex of T sees a_1, a_2, a_4, a_5. Now a_1, a_2, a_4, a_5 are all uncolored vertices that have no neighbor colored c and have at least as many neighbors in A as x_c^{i+1}, so they have the maximum number of neighbors in A, and according to the ordering σ we have $x_c^{i+1} < \min\{a_1, a_2, a_4, a_5\}$. By Theorem 3.2, x_c^{i+1} is not the top of a house, a contradiction. Thus the claim holds. ◻

Claim 2 implies that in the sequence $G = G_1^1, \ldots, G_\ell^{k_\ell}$, each graph other than the first one is obtained from its predecessor by contracting an even pair of the predecessor. Then Lemma 5.1 applied successively along the sequence implies that $\omega(G) = \omega(G_\ell^{k_\ell})$ and $\chi(G) = \chi(G_\ell^{k_\ell})$; but $\chi(G_\ell^{k_\ell}) = \omega(G_\ell^{k_\ell}) = \ell$ since $G_\ell^{k_\ell}$ is a clique of size ℓ; so the algorithm does color the input graph optimally with $\omega(G)$ colors. This completes the proof of the theorem. ◻

Coloring a graph in $\mathcal{B}$ takes time $\mathcal{O}(nm)$ since algorithm LEXBFS* applied on $\overline{G}$ has complexity $\mathcal{O}(nm)$ and Algorithm COSINE* too.

6. Finding a maximum clique. We can extend the preceding algorithms by another greedy algorithm, which, in the case of a graph in class $\mathcal{B}$, will produce in linear time a clique of maximum size. Let G be any graph given with a coloring of its vertices using ℓ colors. Then we can apply the following algorithm to build a set Q:

ALGORITHM CLIQUE
Input: A graph G and a coloring of its vertices using ℓ colors.
Output: A set Q that consists of ℓ vertices of G.
Initialization: Set $Q := \emptyset$, $c := \ell$, and for every vertex x set $q(x) := 0$;
General step: While $c \neq 0$ do:
Pick a vertex x of color c that maximizes $q(x)$, do $Q := Q \cup \{x\}$, for

every neighbor y of x do $q(y) := q(y) + 1$, and do $c := c - 1$.

Algorithm Clique can be implemented in time $\mathcal{O}(m + n)$. To do this, at the step where the vertices of color c are examined, keep one vertex of color c that maximizes the counter q, and update the counter of the neighbors of that vertex.

We claim that when the input consists of a graph G in class $\mathcal{B}$, with the coloring produced by Algorithm LexBFS* followed by Algorithm Cosine*, the output Q of Algorithm Clique is a clique of size ℓ. Actually this will be true in a more general framework.

Lemma 6.1. *Let G be a graph given with a coloring of its vertices using ℓ colors. Call its vertices $x_1^1, x_1^2, \dots, x_1^{k_1}, x_2^1, \dots, x_2^{k_2}, \dots, x_\ell^1, \dots, x_\ell^{k_\ell}$, so that vertices of subscript c have color c. Define the corresponding sequence of graphs G_c^i and vertices w_c^i ($1 \le c \le \ell$, $1 \le i \le k_c$) obtained by successive contractions as in the preceding section. Suppose that for each color $c = 1, \dots, \ell - 1$, we have the following:*

(i) *Every vertex of color strictly greater than c has a neighbor of color c.*

(ii) *For each $i = 1, \dots, k_c - 1$, the graph G_c^i contains no chordless path on four vertices whose endvertices are w_c^i and x_c^{i+1}.*

Let Q be a clique whose vertices have colors strictly greater than c for some $c \in \{1, \dots, \ell - 1\}$. Then there is a vertex of color c that is adjacent to all of Q.

Proof. For $i = 1, \dots, k_c$, consider the following Property P_i: "In the graph G_c^i, vertex w_c^i is adjacent to all of Q." Note that Property P_{k_c} holds by (i) and by the definition of $w_c^{k_c}$. We may assume that Property P_1 does not hold, for otherwise the lemma holds with vertex $x_c^1 = w_c^1$. So there is an integer $i \in \{2, \dots, k_c\}$ such that P_i holds and P_{i-1} does not. Then, in the graph G_c^{i-1}, vertex x_c^i must be adjacent to all of Q, for otherwise Q contains vertices a, b such that a is adjacent to w_c^{i-1} and not to x_c^i and b is adjacent to x_c^i and not to w_c^{i-1}, and then the path w_c^{i-1}-a-b-x_c^i contradicts (ii). So the lemma holds with vertex x_c^i. ∎

Lemma 6.2. *Let G be a graph in class $\mathcal{B}$, and let $x_1^1, x_1^2, \dots, x_1^{k_1}, x_2^1, \dots, x_2^{k_2}, \dots, x_\ell^1, \dots, x_\ell^{k_\ell}$ be a coloring produced by Algorithm* LexBFS* *applied on $\overline{G}$ followed by Algorithm* Cosine* *applied on G. Then, when Algorithm* Clique *is run on this input it produces a clique of size $\omega(G)$.*

Proof. Consider the set Q maintained during Algorithm Clique. We claim that, for each $c = \ell, \ell - 1, \dots, 1$, at the end of step c the set Q is a clique of size $\ell - c + 1$ that contains one vertex of each color $c, \dots, \ell$. This is clear when $c = \ell$. At the general step, Lemma 6.1 ensures that there exists a vertex of color $c - 1$ that is adjacent to all of Q. So Algorithm Clique will select such a vertex, add it to Q, and so the claim remains true at the end of that step. Thus the algorithm ends with a clique Q of size ℓ. Since G admits a coloring of size ℓ, we have $\ell = \chi(G) = \omega(G)$. ∎

7. Comments. We observe that the hypothesis of Lemma 6.2 actually yields some slightly stronger properties:

(a) For any color c, every vertex of color c lies in a clique of size c; and more generally, every clique whose smallest color is c is included in a clique that contains a vertex of each color $1, \dots, c$. This is a consequence of Lemma 6.1 that can be derived just like Lemma 6.2. A coloring that has this property is called *strongly canonical* in [22].

(b) The set of vertices of color 1 is a stable set that intersects all maximal cliques of G. This too can be derived easily from Lemma 6.1. Such a set is called a *strong stable set* in [23]. Thus every graph G in class $\mathcal{B}$ is *strongly perfect* (i.e., every induced subgraph of G has a strong stable set), which was also a corollary of Hayward's result [18]. Moreover, using for graphs in $\mathcal{B}$ the idea from Hoàng [24, Theorem 2.1], this implies that one can find a minimum weighted coloring and a maximum weighted clique for a graph in $\mathcal{B}$ in time $O(n^2m)$.

The coloring algorithm is "robust" [30] in the sense that the input graph can be any graph G, and if G is not in $\mathcal{B}$ and the output coloring is not optimal, it can detect this fault. To do this we apply Algorithm LexBFS* on $\overline{G}$ followed by Algorithm Cosine* and Algorithm Clique on G, and we need only check whether Q is a clique (which can be done in linear time). If Q is a clique, then the coloring is optimal since it uses ℓ colors and Q has size ℓ. If Q is not a clique, then we know that the input graph is not in $\mathcal{B}$.

Since every graph in $\mathcal{B}$ admits a perfect ordering, as proved in [18], one may wonder whether the ordering in which the vertices are colored by Algorithm LexBFS* applied on $\overline{G}$ followed by Algorithm Cosine* applied on G gives such a perfect order. But here is a counterexample. Let G be the graph on six vertices a, b, c, d, e, f, where a-b-c-d-e is a path on five vertices and f is adjacent to a, c, d, e. Then Algorithm LexBFS* applied on $\overline{G}$ can produce the ordering $f < b < c < e < d < a$ and Algorithm Cosine* can color the vertices in the ordering $f < b < c < e < a < d$. This is not a perfect ordering for G since the four vertices b, c, d, e form an "obstruction" [6] since $b < c$ and $e < d$.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Menlo Park, CA, 1974.

[2] C. Berge, *Les problèmes de coloration en théorie des graphes*, Publ. Inst. Statist. Univ. Paris, 9 (1960), pp. 123–160.

[3] C. Berge, *Graphs*, 2nd ed., North-Holland, Amsterdam, New York, 1985.

[4] M. E. Bertschi, *Perfectly contractile graphs*, J. Combin. Theory B, 50 (1990), pp. 222–230.

[5] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, *The strong perfect graph theorem*, Ann. Math. (2), 164 (2006), pp. 51–229.

[6] V. Chvátal, *Perfectly ordered graphs*, in Topics on Perfect Graphs, C. Berge and V. Chvátal, eds., North-Holland Math. Stud., 88, North-Holland, Amsterdam, New York, 1984, pp. 63–65.

[7] V. Chvátal, *A Class of Perfectly Orderable Graphs*, Rpt. 89573-OR, Forschungsinstitut für Diskrete Mathematik, Bonn, 1989.

[8] V. Chvátal and N. Sbihi, *Bull-free Berge graphs are perfect*, Graphs Combin., 3 (1987), pp. 127–139.

[9] C. M. H. de Figueiredo and F. Maffray, *Optimizing bull-free perfect graphs*, SIAM J. Discrete Math., 18 (2004), pp. 226–240.

[10] C. M. H. de Figueiredo, F. Maffray, and O. Porto, *On the structure of bull-free perfect graphs*, Graphs Combin., 13 (1997), pp. 31–55.

[11] C. M. H. de Figueiredo, F. Maffray, and O. Porto, *On the structure of bull-free perfect graphs*, 2: *The weakly chordal case*, Graphs Combin., 17 (2001), pp. 435–456.

[12] H. Everett, C. M. H. de Figueiredo, C. Linhares Sales, F. Maffray, O. Porto, and B. A. Reed, *Even pairs*, in Perfect Graphs, J. L. Ramírez-Alfonsín and B. A. Reed, eds., Wiley Interscience, Chichester, UK, 2001, pp. 67–92.

[13] J. Fonlupt and J. P. Uhry, *Transformations that preserve perfectness and h-perfectness of graphs*, Ann. Discrete Math., 16 (1982), pp. 83–95.

[14] T. Gallai, *Transitiv Orientierbare Graphen*, Acta Math. Acad. Sci. Hungar., 18 (1967), pp. 25–66.

[15] M. Gröstchel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[16] M. Habib, F. de Montgolfier, and C. Paul, *A simple linear-time modular decomposition algorithm for graphs, using order extension*, in Proceedings of SWAT 2004, Lecture Notes in Comput. Sci. 3111, 2004, pp. 187–198.

[17] R. B. Hayward, *Weakly triangulated graphs*, J. Combin. Theory B, 39 (1985), pp. 200–208.

[18] R. B. Hayward, *Bull-free weakly chordal perfectly orderable graphs*, Graphs Combin., 17 (2000), pp. 479–500.

[19] R. B. Hayward, J. Spinrad, and R. Sritharan, *Weakly chordal graph algorithms via handles*, in Proceedings of 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, ACM Press, New York, 2000, pp. 42–49.

[20] A. Hertz, *A fast algorithm for coloring Meyniel graphs*, J. Combin. Theory B, 50 (1990), pp. 231–240.

[21] A. Hertz, C*osine*: *A new graph coloring algorithm*, Oper. Res. Lett., 10 (1991), pp. 411–415.

[22] A. Hertz and D. de Werra, *Connected sequential colorings*, Discrete Math., 74 (1989), pp. 51–59.

[23] C. T. Hoàng, *On a conjecture of Meyniel*, J. Combin. Theory B, 42 (1987), pp. 302–312.

[24] C. T. Hoàng, *Efficient algorithms for minimum weighted colouring of some classes of perfect graphs*, Discrete Appl. Math., 55 (1994), pp. 133–143.

[25] B. Lévêque, F. Maffray, B. A. Reed, and N. Trotignon, *Coloring Artemis Graphs*, Res. Rep. 135, Leibniz Laboratory, Grenoble, France, 2005.

[26] A. Lubiw, *Doubly lexical ordering of matrices*, SIAM J. Comput., 16 (1987), pp. 854–879.

[27] C. Linares Sales and B. A. Reed, eds., *Recent Advances in Algorithms and Combinatorics,* CMS Books Math./Ovurages Math. SMC 11, Springer-Verlag, New York, 2003.

[28] F. Maffray and M. Preissmann, *A translation of Tibor Gallai's paper: Transitiv orientierbare Graphen*, in Perfect Graphs, J. L. Ramírez-Alfonsín and B. A. Reed, eds., Wiley Interscience, Chichester, UK, 2001, pp. 25–66.

[29] H. Meyniel, *A new property of critical imperfect graphs and some consequences*, European J. Combin., 8 (1987), pp. 313–316.

[30] V. Raghavan and J. Spinrad, *Robust algorithms for restricted domains*, J. Algorithms, 48 (2003), pp. 160–172.

[31] B. A. Reed and N. Sbihi, *Recognizing bull-free perfect graphs*, Graphs Combin., 11 (1995), pp. 171–178.

[32] D. J. Rose, R. E. Tarjan, and G. S. Lueker, *Algorithmic aspects of vertex elimination on graphs*, SIAM J. Comput., 5 (1976), pp. 266–283.