SIAM J. DISC. MATH.
Vol. 7, No. 1, pp. 72–89, February 1994

008

ON-LINE COLORING AND RECURSIVE GRAPH THEORY*

H. A. KIERSTEAD†‡, S. G. PENRICE†‡, AND W. T. TROTTER†§

Abstract. An on-line vertex coloring algorithm receives the vertices of a graph in some externally determined order, and, whenever a new vertex is presented, the algorithm also learns to which of the previously presented vertices the new vertex is adjacent. As each vertex is received, the algorithm must make an irrevocable choice of a color to assign the new vertex, and it makes this choice without knowledge of future vertices. A class of graphs Γ is said to be on-line χ-bounded if there exists an on-line algorithm A and a function f such that A uses at most $f(\omega(G))$ colors to properly color any graph G in Γ. If H is a graph, let Forb(H) denote the class of graphs that do not induce H. The goal of this paper is to establish that Forb(T) is on-line χ-bounded for every radius-2 tree T. As a corollary, the authors answer a question of Schmerl's; the authors show that every recursive cocomparability graph can be recursively colored with a number of colors that depends only on its clique number.

Key words. on-line algorithm, graph coloring, recursive function

AMS subject classification. 05C15

1. Introduction. The main result of this article can be formulated in terms of recursive function theory or on-line algorithms. Since the on-line formulation gives a slightly stronger statement and is more universally accessible, we adopt it. However, since the roots of the subject lie in recursive graph theory, we begin with a brief summary of results in this area. A *recursive graph* is a countable graph $G = (V, E)$ such that there exist algorithms (Turing machines) for computing the characteristic functions of V and E. A recursive graph is *highly recursive* if each vertex has finite degree and there exists an algorithm for calculating the degree of each vertex. A graph is *recursively k-colorable* if there exists an algorithm that computes a proper k-coloring of the vertices of the graph. The *recursive chromatic number* of a graph is the least k such that the graph is recursively k-colorable. During the 1970s, several authors, including Manaster and Rosenstein [MR] and Bean, Schmerl, and Kierstead studied the recursive chromatic number of various classes of graphs. For example, Bean [B] proved that every planar highly recursive graph has recursive chromatic number at most 6. Schmerl [S1] showed that every highly recursive k-colorable graph can be recursively $2k$-1-colored, and, in general, this is best possible. He also proved [S2] that Brooks's bound on the chromatic number of a graph also holds for the recursive chromatic number of a highly recursive graph. Kierstead [K2] proved that the recursive chromatic number of a highly recursive perfect graph was, at most, 1 more than its chromatic number and that the recursive edge chromatic number of a highly recursive graph was, at most, 1 more than its edge chromatic number.

While there were many positive results for highly recursive graphs, the results for recursive graphs were almost always negative, unless the class of graphs under consideration had bounded degree. For example, Bean [B] showed that there are recursive forests whose recursive chromatic number is infinite. However, Kierstead [K1] did give a positive result for a similar problem. He answered a question of Schmerl, by showing that every recursive ordered set with width w could be partitioned into at most $(5^w - 1)/4$ recursive chains. From a purely graph theoretical point of view, partitioning

* Received by January 13, 1992; accepted for publication (in revised form) February 9, 1993.
† Department of Mathematics, Arizona State University, Tempe, Arizona 85287.
‡ This author was partially supported by Office of Naval Research grant N00014-90-J-1206.
§ This author was partially supported by National Science Foundation grant DMS-87-13994.

an ordered set of width w into θ chains is equivalent to partitioning a comparability graph with independence number w into θ complete subgraphs, which, in turn, is equivalent to properly coloring a cocomparability graph with clique number w using θ colors. However, Kierstead's algorithm made explicit use of the orientation of the recursive ordered set. These considerations led Schmerl to ask whether there exists a function $f(w)$ such that every recursive comparability graph with independence number w can be partitioned into $f(w)$ recursive cliques. Kierstead and Trotter [KT1] showed that this was true for comparability graphs of interval orders.

Now we consider Schmerl's question from the point of view of on-line algorithms. An *on-line graph* is a structure $G^< = (V, E, <)$, where $G = (V, E)$ is a graph, V is finite or countably infinite, and $<$ is a linear ordering of V. (If V is infinite, then $<$ has the order type of the natural numbers.) We call $G^<$ an *on-line presentation* of the graph G. The on-line subgraph of $G^<$ induced by a subset $X \subset V$ is the on-line graph $G^<[X] = (X, E', <')$, where E' is the set of edges in E both of whose endpoints are in X and $<'$ is $<$ restricted to X. Let $V_i = \{x_1, \ldots, x_i\}$ denote the first i vertices of V in the linear order $<$ and set $G_i^< = G^<[V_i]$. More generally, we refer to on-line structures $S^<$, where S is some structure such as an ordered set or partitioned graph. An algorithm for coloring the vertices of an on-line graph $G^<$ (or, more generally, calculating some function on the universe of an on-line structure) is said to be *on-line* if the color of a vertex x_i is determined solely by $G_i^<$. Intuitively, the algorithm colors the vertices of $G^<$ one at a time in some externally determined order $x_1, \ldots, x_n$, and, at the time a color is irrevocably assigned to the vertex x_i, the algorithm can only see $G_i^<$. A simple but important example of an on-line algorithm is the algorithm First-Fit, denoted by FF, which colors the vertices of G with an initial sequence of the colors $\{1, 2, \ldots\}$ by assigning to the vertex x_i the least possible color not already assigned to any vertex $x \in V_{i-1}$ such that x is adjacent to x_i.

Usually, an algorithm for recursively coloring recursive graphs results in an on-line algorithm, while more specialized algorithms for coloring highly recursive graphs do not. The reason for this is that algorithms for coloring highly recursive graphs can learn about the neighbors of a vertex, or the neighbors of the neighbors of a vertex, and so forth, before coloring the vertex. An on-line algorithm for coloring graphs always produces an algorithm for coloring recursive graphs. In this vein, the proof of Kierstead's recursive chain covering theorem actually yields the following slightly stronger statement.

THEOREM 1.1. *There exists an on-line algorithm A that will partition any on-line ordered set $P^<$ into $(5^w - 1)/4$ chains.*

Similarly, Bean's example of a forest with infinite recursive chromatic number produces an on-line tree that cannot be finitely colored by any on-line algorithm.

Schmerl's question proves to be a special case of a more general problem. Before continuing, we introduce some terminology and graph theoretical results. The *clique size* and *chromatic number* of a graph G are denoted by $\omega(G)$ and $\chi(G)$, respectively. Let A be an on-line graph coloring algorithm. Then $\chi_A(G^<)$ denotes the number of colors A uses to color the on-line graph $G^<$, and $\chi_A(G)$ denotes the maximum of $\chi_A(G^<)$ over all on-line presentations $G^<$ of G. A class of graphs Γ is said to be *χ-bounded* if there exists a function f such that, for all $G \in \Gamma$, $\chi(G) \le f(\omega(G))$. Easy examples of χ-bounded classes include the class of perfect graphs (which include cocomparability graphs), the class of line graphs, and, more generally, the class of claw-free graphs. Similarly, for an on-line algorithm A, the class Γ is *χ_A-bounded* if there exists a function f such that, for all $G \in \Gamma$, $\chi_A(G) \le f(\omega(G))$. The class Γ is *on-line χ-bounded* if Γ is χ_A-bounded for some on-line algorithm A. The class of perfect graphs is not on-line χ-bounded. In fact, the subclass of trees is not even on-line χ-bounded as we noted above. However, the class of claw-free graphs is on-line χ-bounded. We now rephrase Schmerl's question in these terms.

Question 1.2. Is the class of cocomparability graphs on-line χ-bounded?

A graph $H = (X, F)$ is an *induced subgraph* of a graph $G = (V, E)$ if and only if (1) $X \subset V$, and (2) for all vertices $x, y \in X$, $xy \in F$ if and only if $xy \in E$. For a graph H, let Forb(H) be the class of graphs G such that H is not isomorphic to an induced subgraph of G. In the mid-1970s, Gyárfás [G1] and Sumner [Su] independently formulated the following conjecture.

CONJECTURE 1.3. *For any tree T, the class of graphs* Forb(T) *is χ-bounded.*

Several comments about this conjecture are in order. First, it is easy to show (see [G1] or [Su]) that, if $\chi(G) = k$, then G contains every tree T on k vertices as a subgraph, but not necessarily as an induced subgraph. Second, if Forb(H) is χ-bounded, then H is acyclic. This follows immediately from a result of Erdös and Hajnal [EH] that, for every positive integer $i > 2$, there exists a graph G_i such that both the girth and chromatic number of G_i are at least i. Such graphs have clique number 2 and do not contain any graph that contains a cycle of length i. Finally, if F is a forest, Forb(F) is χ-bounded if and only if, for each of the connected components T_i of F, the class Forb(T_i) is χ-bounded. Thus, if the conjecture is true, its proof yields a characterization of those graphs H such that Forb(H) is χ-bounded.

Rödl proved a weaker version of the conjecture. He showed [KR] that, for every tree T and complete bipartite graph $K_{t,t}$, the class Forb(T) $\cap$ Forb($K_{t,t}$) is on-line χ-bounded. Gyárfás [G2] showed that the conjecture is true when T is any path. Gyárfás, Szemerédi, and Tuza [GST] verified the conjecture for triangle-free graphs in Forb(T), when T is any radius-2 tree, and Kierstead and Penrice [KP1] extended this result by showing that Forb(T) is χ-bounded whenever T has radius 2. The latter two results use Rödl's theorem. We need the following strengthening of Rödl's result due to Kierstead and Penrice [KP1].

THEOREM 1.4. *For every tree T and complete bipartite graph $K_{t,t}$,* Forb(T) $\cap$ Forb($K_{t,t}$) *is χ_{FF}-bounded.*

An old result of Chvátal [C] shows that Forb(P_4) is ψ_{FF}-bounded, where P_n is a path on n vertices. Gyárfás and Lehel [GL3] made an exciting and unexpected breakthrough when they extended this result by proving that Forb(P_5) is on-line χ-bounded. They also showed that Forb(P_6) is not on-line χ-bounded. Thus, if Forb(T) is on-line χ-bounded for some tree T, then T has radius at most 2. The central result of this article is that this condition is not only necessary, but is also sufficient.

THEOREM 1.5. *For every tree T, the class* Forb(T) *is on-line χ-bounded if and only if T is a radius-2 tree.*

We are indebted to Gyárfás for reminding us that, as a consequence of Gallai's characterization of comparability graphs [Ga], cocomparability graphs do not induce the radius-2 tree obtained by subdividing each edge of $K_{1,3}$ (see Fig. 1.1). Of course, this is part of the easy direction of Gallai's characterization and can be readily verified from scratch. Thus, as an immediate corollary, we obtain the following answer to Schmerl's question.

COROLLARY 1.6. *The class of cocomparability graphs is on-line χ-bounded.*

This paper is organized as follows. In the remainder of this section, we state some preliminary results and review our notation and terminology. In §2 we give an overview of the off-line proof and the problems we must deal with to create an on-line algorithm. In §§4 and 5 we develop some purely combinatorial lemmas needed to verify the correctness of the main algorithm. In §5 we also present a key on-line subroutine and in §6

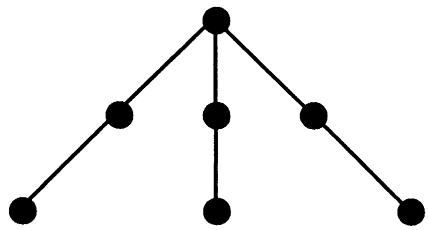

FIG. 1.1.

we give the proof of Theorem 1.5, after giving a more technical reformulation of it as Theorem 6.1.

We use the following easy lemma, which follows, for example, from Turan's theorem.

LEMMA 1.7. *Let D be a directed graph, where Δ^{out} denotes the maximum outdegree of D and where ν and ε denote the number of vertices and edges of D, respectively. Then D contains an independent set of size at least $\nu/(2\Delta^{\text{out}}+1)$.*

Let $R[a, b]$ denote the Ramsey function with the property that every graph on $R[a, b]$ vertices contains an independent set of size a or a complete subgraph of size b.

Let $T_{a,b}$ be the radius-2 tree whose root r is adjacent to a level-1 vertices $x_1, \dots, x_a$, and each level-1 vertex x_i is adjacent to b leaves $y_{i,1}, \dots, y_{i,b}$. We call the set $\{y_{i,1}, \dots, y_{i,b}\}$ a level-2 group. We abbreviate $T_{k,k}$ by T_k. The complete s-partite graph with w vertices in each part is denoted by K_w^s.

If two vertices x and y are adjacent, we write $x \sim y$. If X and Y are sets of vertices such that every vertex in X is adjacent to every vertex in Y, then we write $X \sim Y$. We denote the neighborhood of a vertex x by $N(x) = \{y: x \sim y\}$.

2. Overview. In this section, we give an overview of the proof of the off-line version of our main theorem and the additional problems that must be solved to prove the on-line theorem. This section also serves as a guide to reading the rest of the paper. We begin by noting that, if T' is a subtree of the tree T, then $\text{Forb}(T') \subset \text{Forb}(T)$, and thus $\text{Forb}(T')$ is on-line χ-bounded if $\text{Forb}(T)$ is. Thus it suffices to prove that, for all k, $\text{Forb}(T_k)$ is on-line χ-bounded. In §3, two elementary combinatorial lemmas on trees are presented, which simplify the remaining arguments. In particular, quasi-induced trees are defined, and it is shown that it suffices to prove that the smaller class $q\,\text{Forb}(T_k)$ of graphs that do not contain a quasi-induced T_k is on-line χ-bounded.

The central idea in both the on-line and off-line proofs is the notion of s-templates and their use to partition the graph so that the vertices can be properly colored in terms of local and global colors. Roughly, an s-template is a complete s-partite graph with a very large number of vertices in each part. The exact definition of its part size depends on $t = \omega(G)$ and s. However, there will be an absolute upper bound on part size in terms of t. At a given stage of a double induction on t and s, we assume the following for some bound c depending on s and t:

(1) If H is a graph on $q\,\text{Forb}(T_k)$ such that either (a) $\omega(H) < t$ or (b) $\omega(H) = t$ and H does not contain an induced s-template, then $\chi(H) \le c$.

It is easy to partition the vertices of G as $(X, B_1, O_1, B_2, O_2, \dots, B_n, O_n)$ so that, for $1 \le i < j \le n$, the following assumption holds:

(2) (a) B_i is an s-template,

(b) Each vertex in O_i is adjacent to at least k vertices in some part of B_i,

(c) No vertex in B_j is adjacent to k or more vertices in any part of B_i,

(d) X does not contain an s-template.

By part (b) of assumption (1), we can color X with c colors. Using a different disjoint set of $c|B_i|$ colors, we can, by part (a), color each O_i with $c|B_i|$ colors so that all vertices of O_i that receive the same color have a common neighbor in B_i. However, two adjacent vertices, one in O_i and the other in O_j with $i \neq j$, may receive the same color. Finally, using a third disjoint set of $|B_i|$ colors, we can color each B_i so that each vertex of B_i receives a distinct color. Again, two adjacent vertices, one in B_i and the other in B_j with $i \neq j$, may receive the same color. Call these colors local colors and let $\langle \alpha \rangle$ denote the set of vertices with local color α.

It remains to show that we can color each local color class $\langle \alpha \rangle$ with a bounded number of colors. Before describing this coloring, we must mention one technical complication. A point x is said to be an extra point for an s-template B_i if, roughly, x is adjacent to almost every vertex in every part of B_i. Extra points will create all sorts of minor problems, which require special attention. Fortunately, if any template has too many extra points, we will be able to start over using an algorithm based on $(s+1)$-templates. For the rest of this informal discussion, we ignore the possibility of extra points. With this sluff, we can give a simple statement of the following crucial properties of our partition. In §4 these properties are stated in full technical detail and proved.

(3) There exists a constant d such that, for any vertex x, local color α, and integer $i < n$,

(a) $|\{j : x \sim y, \text{ for some } y \in B_j\}| \leq d$,

(b) $|\{j : x \text{ is adjacent to at least } k \text{ vertices in } O_j \cap \langle \alpha \rangle\}| \leq d$,

(c) $|\{j : \text{for some } x \in O_i \cap \langle \alpha \rangle,\ x \text{ is adjacent to at least } k \text{ vertices in } O_j \cap \langle \alpha \rangle\}| \leq d$.

Properties (a)–(c) in the above assumption allow us to color each $\langle \alpha \rangle$ with a bounded number of colors as follows. If α is a local color that is used on a vertex in some B_i, then α is used on exactly one vertex of each B_j. Thus, by part (a), the degree of $\langle \alpha \rangle$ is bounded by d, and so $\langle \alpha \rangle$ can be $d+1$ colored. Suppose that α is a local color that is used on a vertex of O_i. We define a directed auxiliary graph G' on the vertices $\{1, \ldots, n\}$ by $i \to j$ if and only if there exists a vertex $x \in O_i \cap \langle \alpha \rangle$ such that x is adjacent to at least k vertices of $O_j \cap \langle \alpha \rangle$. By part (c), G' has outdegree at most d and thus can be colored with at most $2d+1$ colors. We assign each vertex $x \in \langle \alpha \rangle$ a two-coordinate global color. The first coordinate is the color of i in G', if $x \in O_i$. Now let $\langle \alpha, \beta \rangle$ be the subset of $\langle \alpha \rangle$ of vertices whose global color has first coordinate β. By (a) and (b), the degree of $\langle \alpha, \beta \rangle$ is less than d^2, and thus $\langle \alpha, \beta \rangle$ can be colored with d^2 colors. We have thus properly colored G with a bounded number of colors.

Next, we consider the problems involved in implementing the above proof on-line. The major problem is that we cannot calculate the partition of G into $(X, B_1, O_1, B_2, O_2, \ldots, B_n, O_n)$ on-line. When a vertex x is presented, it may appear to belong to the first part X of the partition, but later we may learn that it must be assigned to some B_i or O_i. Without being able to properly assign x to a part of the partition, we have no basis for coloring x. Now suppose that we are very fortunate and that whenever a new vertex x is presented we correctly guess its proper position in the partition. There is still a minor problem. Suppose that x is correctly assigned to O_i. Then x is given a global color, which is based in part on the auxiliary graph G'. However, the presentation of x may cause an edge from i to j to appear in G', when previously we had assigned i and j the same color. This problem is solved in §5, where the definition of the auxiliary graph is slightly modified to facilitate dealing with the extra points. In particular, the auxiliary graph will not be directed.

To handle the more serious problem, our on-line algorithm will maintain a partition of G into $P = (X, B_1, O_1, B_2, O_2, \ldots)$, which approximates the desired partition in the following sense. At any stage $i+1$, when we consider the vertex x_{i+1}, P will be a partition of G_i, which satisfies parts (a)–(d) of assumption (2). At any stage, vertices may be removed from X to form B_{j+1}, where B_j is the last template in the previous partition. When this happens, other vertices may be moved from X to O_i. Once a vertex is assigned to a part of the partition other than X, it will never move. Thus, when we are presented with x_{i+1}, we try to assign x_{i+1} to some O_j. If this is not possible, we try to form a new s-template with x_{i+1}, together with some vertices from X. If this is not possible, we assign x_{i+1} to X.

We are left with the problem of coloring the newly presented vertex x_{i+1}. There is no problem if x_{i+1} is assigned to a new s-template B_j, and, if x is assigned to some O_j, it is relatively easy to color x using the techniques of §5, referred to above. The main problem arises when x_{i+1} is first assigned to X. In this case, we do not know where x_{i+1} will end up, so we must somehow hedge our bets. We would like to color x_{i+1} using the on-line version of part (b) of (1). The set of points currently in X does not contain an s-template, but this property is maintained artificially by removing vertices that would otherwise form an s-template in X. Simply removing vertices from X does not solely solve the problem of coloring vertices of X', the set of points originally assigned to X, because the color of a vertex originally assigned to X continues to influence future vertices even after the point is removed from X. We cannot afford to change the set of colors used for vertices entering X every time vertices are removed from X, because we may have to change color sets an unbounded number of times due to the fact that the template sequence may be unbounded in length. Thus, we want part (b) of (1) to forbid, not s-templates in X, but much larger s-partite graphs K_w^s in X', and we will be able to do so because of the other properties of the algorithm. By parts (a) and (b) of assumption (3), the vertices of a supposed K_w^s in X' could not have been used to form too many different s-templates, nor could this set of vertices intersect too many of the O_j's. Thus, if X' contains a copy B' of K_w^s, then, by setting w large enough, we can assume that, for each part Q_a of B', there exists a large subset Q'_a such that either $Q'_a \subset X$ or $Q'_a \subset O_{j(a)} \cup B_{j(a)}$ for some $j(a)$. Here, large means the size of a part in an s-template. This motivates the intricate induction hypothesis presented in §6, where we will color X' on-line with a two-coordinate color. The first coordinate will ensure that, if $x, y \in X'$, $x \sim y$, and $x \in X' - X$ at the time y is presented, then x and y receive different colors. From this fact, following the remarks above, we will show that no first coordinate color class of X' contains K_w^s for appropriately chosen w. Thus we will be able to color each of these first coordinate color classes on-line by a revised version of the induction hypothesis (b) of (1).

3. Lemmas on radius-2 trees. In this section, we develop some preliminary results about trees. We begin with some fundamental definitions. A graph H is called a *pseudo-induced* $T_{a,b}$ if H has a spanning tree T that is isomorphic to $T_{a,b}$ and the root of T is not adjacent in H to any leaf of T. A graph H is called a *quasi-induced* T_k if H has a spanning tree T that is isomorphic to T_k, and, if xy is an edge in H that is not present in T_k, then x and y are either both level-1 vertices, or they are both level-2 vertices. Note that quasi-induced T_k is the stronger of the two: Every quasi-induced T_k is a pseudo-induced T_k, but a pseudo-induced T_k may have "extra" edges between the first and second levels. In a similar vein, we say that H is an *augmented* K_w^s if $V(H)$ can be partitioned into s sets $Q_1, \ldots, Q_s$ of size w such that $x \sim y$ whenever $x \in Q_i$, $y \in Q_j$, and $1 \le i < j \le s$. We abuse standard usage by calling the Q_i's "parts" of H, even though they need not be independent sets.

We introduced the definition of a quasi-induced T_k to simplify future arguments. It is easier to verify that a graph contains a quasi-induced T_k than it is to verify that a graph contains an induced T_k, because, in the latter case, we must in effect check every pair of vertices to see if the proper edge or nonedge is present, whereas, with quasi-induced trees, we need only check pairs of vertices from different levels. The next lemma shows that the results we desire for graphs that do not contain induced trees can be obtained from results about graphs that do not contain quasi-induced trees.

LEMMA 3.1. *For all positive integers k and t, there exists a positive integer $k'(k, t)$ such that, if $k' = k'(k, t)$ and H is a quasi-induced $T_{k'}$, then H contains either an induced T_k or a clique on t vertices.*

Proof. If k' is sufficiently large, then, by repeated applications of the Bipartite Ramsey Theorem, we may assume that, between any two level-2 groups, either all edges are present, or no edges are present. If there are $R[R[k, t], t]$ level-2 groups, then either there exist t level-2 groups with all edges between any two distinct groups present, in which case there exists a t-clique in H, or there exist $R[k, t]$ level-2 groups with no edges present between any two groups. By Ramsey's theorem, among the level-1 vertices associated with each of these $R[k, t]$ level-2 groups, there exists either a t-clique or a set of k independent vertices. In the latter case, however, we have an induced T_k. □

We use q Forb(T_k) to denote the class of graphs that do not contain a quasi-induced T_k. Lemma 3.1 may then be paraphrased as "If k' is sufficiently large, then q Forb($T_{k'}$) $\supseteq$ Forb(T_k)." Henceforth, our arguments will refer to quasi-induced trees rather than induced trees. Also, we denote by q Forb(K_w^s) the class of graphs that do not contain an augmented K_w^s, and q Forb(T_k, K_w^s) denotes the class of graphs that contain neither a quasi-induced T_k nor an augmented K_w^s.

LEMMA 3.2. *Let $H \in q$ Forb(T_k) be a graph that is spanned by T, a pseudo-induced $T_{k(2a+1),kb}$. Then some level-1 vertex x of T has b neighbors in a distinct level-2 group, other than the level-2 group associated with x.*

Proof. Let $x_1, \ldots, x_{k(2a+1)}$ be the level-1 vertices of T and let $S_1, \ldots, S_{k(2a+1)}$ be the level-2 groups of T. Suppose that no level-1 vertex has b neighbors in a distinct level-2 groups. Define a directed graph D on the vertex set $\{1, \ldots, k(2a+1)\}$ by directing an arc form i to j, $i \neq j$, if and only if x_i has b neighbors in S_j. By our supposition, $\Delta^{\text{out}}(D) \leq a$. It follows from Lemma 3.2 that D has an independent set of size k. This means that we may assume without loss of generality that, for $i = 1, \ldots, k$, x_i has less than a neighbors in each S_j, $1 \leq j \leq k$, $j \neq i$. Thus, by removing at most $(k-1)(a-1)$ vertices from each S_j, leaving at least k vertices in each level-2 group, we find a quasi-induced T_k, a contradiction. □

4. Definitions and structural lemmas. Let k, p, s, and t be fixed positive integers such that $s \geq 2$ and $p > k^4$. If B is a subgraph of G isomorphic to an augmented K_w^s with $w = p + k^4$, then we call B a *template* of G. We call a vertex $x \notin B$ an *extra point* of B if x has less than k^4 nonneighbors in each part of B. We call a vertex $x \notin B$ a *strong 1-neighbor* of B if x has k neighbors in some part of B. Note that an extra point is also a strong 1-neighbor. A sequence of templates $B_1, \ldots, B_r$ is called an *acceptable template sequence* if, whenever $1 \leq m < n \leq r$ and $x \in B_n$, x is not a strong 1-neighbor of B_m.

This section has two goals: to define an i-tube and to prove a technical lemma, Lemma 4.1, which will play a crucial role when we verify that our on-line algorithm uses a bounded number of colors. In particular, it will help us to verify that the auxiliary graphs have bounded degree, and it will have other applications as well. The following definition and the statement of Lemma 4.1 are, in fact, the only elements of this section used in the rest of the paper. The first-time reader may wish to proceed to §5 after studying the remarks following the statement of Lemma 4.1 and return to the proofs of this section later.

DEFINITION. Let i be a nonnegative integer. Then we call a subgraph $U = (B, N_1, N_2, \ldots, N_i)$ an *i-tube* if

(i) B is a template,
(ii) $B \cap N_m = \varnothing$ for $1 \leq m \leq i$,
(iii) $N_m \cap N_n = \varnothing$ for $1 \leq m < n \leq i$,
(iv) $|N_m| = k$ for $1 \leq m < i$ and $|N_i| = 1$,
(v) If $x \in N_1$, then x has at least k neighbors in some part of B,
(vi) $N_m \sim N_{m+1}$ for $1 \leq m \leq i - 1$.

We refer to the unique element of N_i as the *top* of U. We refer to N_m as the mth *level* of U for $1 \leq m \leq i$. Also, B is the 0th level of U. If U is a 0-tube, then B is the top level.

LEMMA 4.1. *There exists a function $f_1(i, k, \rho)$ such that, for every graph $G \in q$ Forb(T_k) and every vertex x of G, if*

(i) $\mathbf{U} = \{U_1, \ldots, U_j\}$ *is a collection of pairwise disjoint i-tubes in G with* 0*-levels $B_1, \ldots, B_j$, forming an acceptable template sequence,*
(ii) *x is not an extra point of any template in the sequence $B_1, \ldots, B_j$,*
(iii) *x has a neighbor in the top level of each of $U_1, \ldots, U_j$,*
(iv) *B_n has less than ρ extra points for $1 \leq n \leq j$,*

then $j < f_1(i, k, \rho)$.

Lemma 4.1 has simple interpretations in the cases where $i = 0$ and $i = 1$. Namely, if x is a vertex of a graph $G \in q$ Forb(T_k) and there exists an acceptable sequence of templates $B_1, \ldots, B_j$ such that x has a neighbor in B_n but x is not an extra point of B_n for $1 \leq n \leq j$, then $j < f_1(0, k, \rho)$, because a template is, in fact, a 0-tube. Similarly, if, instead of assuming that x has neighbors in the B_n's, we assume that x has neighbors $x_1, \ldots, x_j$ such that x_n is a strong 1-neighbor of B_n for $1 \leq n \leq j$, then $j < f_1(1, k, \rho)$, because $B_n \cup \{x_n\}$ is a 1-tube for $1 \leq n \leq j$. (In the latter case, we retain the assumption that x is not an extra point of the templates; for the x_n's, we need not make a distinction between extra points and nonextra points.) The case where $i = 2$ is also used in our proof, but describing it informally outside the context of the algorithm is awkward.

We surmise that tubes play a role in proving off-line results concerning trees of radius larger than 2; this is the theoretical reason for proving a general version of Lemma 4.1 (for arbitrary i) when only the cases where $i = 0$, 1, and 2 are used. (As a practical matter, the heart of the proof, Lemma 4.3, is proved by induction, so the general result is obtained with more economy than proving these three cases separately.) We also remark that the hypotheses of Lemma 4.1 can be weakened if a corresponding change is made in the definition of an i-tube. Specifically, the tubes need not be completely disjoint, provided that the bases form an acceptable template sequence, which, by definition, consists of disjoint templates. However, to realize this apparent strengthening of the lemma, we must add to the definition of an i-tube the condition that no vertex in the tube is an extra point of the base. After making this adjustment, a different version of Lemma 4.1 enables us to prove Theorem 6.1 using an on-line algorithm, which, while more complicated, appears to use fewer colors than the algorithm of this paper.

Establishing Lemma 4.1 requires some other purely technical lemmas. We state and prove each separately.

LEMMA 4.2. *Let B be a template and suppose that $x \notin B$ is not an extra point of B. If $1 \leq q \leq k^4$ and x has q neighbors in some part of B, then there exist vertices $x_1, \ldots, x_q$ in one part P of B and $y_1, \ldots, y_{k^4}$ in a different part Q such that x is adjacent to x_m for $1 \leq m \leq q$ and x is not adjacent to y_m for $1 \leq m \leq k^4$.*

Proof. Since $p \geq k^4$, every part of B contains either k^4 neighbors of x or k^4 nonneighbors of x, by the pigeonhole principle. Since x is not an extra point of B, then some part, say Q', of B contains k^4 nonneighbors of x. By the hypothesis, some part of B, say P', contains q neighbors of x. If $P' \neq Q'$, set $P = P'$ and $Q = Q'$, and we are done. If $P' = Q'$, then, since $s \geq 2$, we may consider another part, say R. As we observed earlier, either R contains k^4 neighbors of x or R contains k^4 nonneighbors of x. In the former case, set $P = R$ and $Q = Q'$. In the latter case, set $P = P'$ and $Q = R$. It is then easy to find the desired vertices. □

As we indicated at the beginning of this section, one of our short-term goals is to prove Lemma 4.1, which states roughly that there is a bound on the number of tubes in which a point can have neighbors. To prove this bound, it is helpful to prove a bound for a sequence of tubes with special properties and then extend the results to more general sequences of tubes. Thus we are motivated to introduce the following definition. A sequence of i-tubes $U_1, \ldots, U_j$, where $U_n = (B_n, N_{1,n}, \ldots, N_{i,n})$, is called an *acceptable j-sequence of i-tubes* if

(i) $U_m \cap U_n = \varnothing$ for $1 \leq m < n \leq j$,
(ii) The sequence $B_1, \ldots, B_j$ is an acceptable template sequence,
(iii) If $x \in U_m$, then x is not an extra point of B_n for $1 \leq m \neq n \leq j$.

Having introduced this definition, we now show that it is reasonably easy to establish the kinds of bounds we seek when we consider acceptable sequences of i-tubes. The following argument, with weaker bounds, was useful in [GST] and [KP1].

LEMMA 4.3. *Let $\{b_i\}$ be the sequence of functions defined by $b_0(k) = k(2k + 1)$ and $b_{i+1}(k) = k(2b_i(k) + 1) + b_i(k)$ if $i > 0$. Let $\mathrm{U} = \{U_1, \ldots, U_j\}$ be an acceptable j-collection of i-tubes, for some fixed $i \geq 0$, in some graph $G \in q\,\mathrm{Forb}(T_k)$. Suppose that x is a vertex such that x is adjacent to the top vertex of U_n and x is not an extra point of B_n, for $1 \leq n \leq j$. Then $j < b_i(k)$.*

Proof. Let k be fixed and assume for notational ease that $b_i = b_i(k)$ for $i \geq 0$. We induct on i. Let $i = 0$ and suppose for contradiction that x has a neighbor in B_n for $1 \leq n \leq j = b_0$. Our first goal is to find a vertex y, a strictly increasing function σ, and sets $B'_{\sigma(i)}$ and $B''_{\sigma(i)}$ for $i = 1, \ldots, k$ such that

(1) $B'_{\sigma(i)}, B''_{\sigma(i)} \subset B_{\sigma(i)}$,
(2) $|B'_{\sigma(i)}|, |B''_{\sigma(i)}| \geq k^3$,
(3) y is adjacent to every vertex of $B'_{\sigma(i)}$,
(4) y is nonadjacent to every vertex of $B''_{\sigma(i)}$,
(5) $B'_{\sigma(i)} \sim B''_{\sigma(i)}$.

By applying Lemma 4.2 to x and to each of the templates $B_1, \ldots, B_j$, we may find T_1, a pseudo-induced $T_{a,b}$ with $a = b_0$, $b = k^4$, whose level-2 groups are contained in distinct templates. Applying Lemma 3.2, we find a level-1 vertex y of T_1, which has a set $B'_{\sigma(i)}$ of k^3 neighbors in each of k templates $B_{\sigma(1)}, \ldots, B_{\sigma(k)}$, none of which are at the base of the tube containing y. Observing that (by the definition of an acceptable sequence of i-tubes) y is not an extra point of any of the templates, we may apply Lemma 4.2 again to find $B''_{\sigma(1)}, \ldots, B''_{\sigma(k)}$ as desired. Without loss of generality, $\sigma(i) = i$, for all i.

Having found the structures with properties (1)–(5), we now seek to find a quasi-induced T_k. To do so, we construct a sequence of sets $S_1, \ldots, S_k$ and a sequence of vertices $R_1, \ldots, R_k$ as follows. Let R_k be any vertex of B'_k and let S_k be any k-element subset of B''_k. Suppose that $R_k, S_k, R_{k-1}, S_{k-1}, \ldots, R_r, S_r$ are constructed for $r > 1$. Define R_{r-1} to be any vertex of B'_{r-1} that is not adjacent to any vertex of $S_r \cup \cdots \cup S_k$. This is possible, since each vertex w of $S_r \cup \cdots \cup S_k$ has fewer than k neighbors in B'_{r-1} and $|S_r \cup \cdots \cup S_k| < k^2$. Define S_{r-1} to be any k-element subset of B''_{r-1} that does not contain any neighbors of $R_r, \ldots, R_k$. Again, this is possible because each vertex $R_r, \ldots, R_k$ has fewer than k neighbors in B''_{r-1} and $|\{R_r, \ldots, R_k\}| < k$.

When we have chosen $R_1, \ldots, R_k, S_1, \ldots, S_k$, these vertices, together with y, form a quasi-induced T_k.

Suppose that $i > 0$. Without loss of generality, x has a neighbor in each of $N_{i,1}, \ldots, N_{i,b_i}$; say these neighbors are $x_1, \ldots, x_{b_i}$. By induction, we may assume that x has no neighbors in $N_{i-1,1}, \ldots, N_{i-1,b_i-b_{i-1}}$. Then x is the root of a pseudo-induced $T_{a,b}$ with $a = k(2b_{i-1} + 1)$ and $b = k$. Then, by Lemma 3.2, we find a level-1 vertex x_n that has a neighbor in b_{i-1} level-2 groups contained in tubes that do not contain x_n. This is a contradiction of the induction hypothesis, since, by the definition of an acceptable sequence, x_n is not an extra point of any template that is at the base of a tube (from the acceptable sequence) not containing x_n. □

Proof of Lemma 3.1. Let $f_1(i, k, \rho) = (b_i(k))(2\rho + 1)$. Suppose that conditions (i)–(iv) of the hypothesis hold with $j = f_1(i, k, \rho)$. Define a digraph on $\{1, \ldots, j\}$ by directing an arc from m to n if and only if U_n contains an extra point of B_m. Since the U's are pairwise disjoint, the digraph has outdegree at most ρ; by Lemma 1.7, we must find an independent set of size $b_i(k)$ in the digraph. However, independent sets in this digraph are acceptable tube sequences in G. Thus, we may assume that $\mathbf{U}' = \{U_1, \ldots, U_b\}$ is an acceptable b-sequence of i-tubes, where $b = b_i(k)$. Then, since x has a neighbor in the top of each of these tubes, Lemma 4.3 implies that $b_i(k) < b_i(k)$, a clear contradiction. □

5. Lemma for using auxiliary graphs on-line. We now deal with the minor problem mentioned in our overview. We wish to show that to color a graph $G \in q$ Forb(T_k) on-line with a bounded number of colors, it suffices to partition G on-line into independent sets $I_1, \ldots, I_r$ with the following properties:

(a) For all m, $1 \le m \le r$, $|\{n: I_m \cup I_n$ contains an augmented $K_{e,e}\}| \le f$,

(b) If $H \cong K_{d,d}$ is a subgraph of G, then there exist I_p and I_q such that some subset of $(i_p \cup I_q) \cap H$ contains an augmented $K_{e,e}$.

The number of colors used will be bounded in terms of d, e, f, and $\omega(G)$. Many of the essential ideas are present in a transparent way when we consider the off-line setting. We begin with this argument.

PROPOSITION 5.1. *There exists a constant c depending only on d, e, f, and $\omega(G)$, such that, if $G \in q$* Forb(T_k) *can be partitioned into independent sets as in* (a) *and* (b), *then $\chi(G) \le c$.*

Proof. Suppose that G is partitioned into independent sets $I_1, \ldots, I_r$ in a way satisfying (a) and (b). Define a graph G' on the sets $I_1, \ldots, I_r$ by declaring that I_m is adjacent to I_n if and only if $I_m \cup I_n$ contains an augmented $K_{e,e}$. Note that, by the definition of (a), $\Delta(G') \le f$, and it easily follows that $\chi(G') \le f + 1$. Let g' be an optimal coloring of G'. Define a two-coordinate coloring g of G as follows. Compute the first coordinate of each vertex x by assigning x the color $g'(I_n)$, where I_n is the independent set in the partition that contains x. After all the first coordinates have been computed, for every $\alpha \in$ range (g'), define a graph G_α as the subgraph of G induced by vertices that received color α in their first coordinate. Let g_α be an optimal coloring of G_α and let $g_\alpha(x)$ be the second coordinate of $g(x)$. It is clear that g is a proper coloring of G and that

$$|\text{range } (g)| \le |\text{range } (g')| \max \{|\text{range } (g_\alpha)|: \alpha \in \text{range } (g)\}.$$

It remains for us to verify that $|\text{range } (g)|$ is bounded in terms of d, e, f, and $\omega(G)$. We have already noted that $|\text{range } (g')| \le f + 1$. Since q Forb(T_k, $K_{d,d}$) is χ-bounded (indeed, it is χ_{FF}-bounded), it suffices to show that $G_\alpha \in q$ Forb($K_{d,d}$). Suppose that G_α contains an augmented $K_{d,d}$; call this subgraph H. By condition (b), there exist independent sets I_p and I_q of the partition such that $(I_p \cup I_q) \cap H$ contains an augmented $K_{e,e}$. Then, however, I_p and I_q are adjacent in G', so vertices on opposite sides of the $K_{d,d}$ received different colors in their first coordinate of g, contradicting the fact that every vertex of H received color α in its first coordinate. □

We refer to G' as the *auxiliary graph*. To avoid confusion with the vertices of G, we call the points of G' *nodes*. There are several adjustments that must be made in the on-line case. The greatest difficulty arises from the fact that, when G and the partition $I_1, \ldots, I_r\}$ are presented on-line in G', it is possible that an edge may be "discovered" in G' well after both of its nodes have appeared. This is because the nodes of the auxiliary graph are sets of vertices of G, and edges are formed in the auxiliary graph only when a large number of edges (of G) are present between the two sets of the partition. Thus the auxiliary graph does not strictly fall under the on-line model. However, as colorers, we are aided by the fact that (again, in contrast to the standard on-line model) we may change the color of a node as we discover new edges incident on the vertex, provided that the number of times we change the color of a node is no larger than the degree of the node.

LEMMA 5.2. *There exists an on-line algorithm A and a constant c depending on d, e, f, and ω such that, if $G^<$ is an on-line presentation of a partitioned graph $G = V, E, I_1, I_2, \ldots)$, with $\omega(G) \leq \omega$ and $G \in q$* Forb(T_k), *satisfying* (a) *and* (b), *then A colors $G^<$ using at most c colors.*

Proof. Let G' be defined as in Proposition 5.1. Without loss of generality, we may assume the node set of G' is the set of positive integers. As a new vertex in G is presented, it may cause a previously unseen edge to appear in G'. Despite this complication, we seek to maintain a proper coloring of G', even though we may occasionally change the color of a node of G'.

Suppose that a vertex x enters G and is assigned to I_m. If x does not produce any new edges incident on I_m, do not change the coloring of G'. If x does produce one or more new edges incident on I_m, assign I_m a new color (if necessary) so that the color of I_m is different from all of its previous colors, as well as the current colors of all the neighbors of I_m in G'. Since I_m has at most f neighbors in G' and each vertex of G' has at most $f + 1$ different colors throughout its history (since only the addition of an incident edge can cause a color change), a color will always be available for I_m, provided that we use $f^2 + f + 1$ colors to color G'. Moreover, we have something more than a proper coloring: After an edge $I_m I_n$ appears, I_m is never given a color previously held by I_n, and vice versa.

Now x will be assigned a two coordinate color. The first coordinate will be the color of I_m in G' after any edges in G' caused by adding x to G have been added to G'. To compute the second coordinate, apply First-Fit to the subgraph of G induced by the vertices that received the same first-coordinate color as x.

Clearly, this algorithm gives a proper coloring of G, and we have already determined that at most $f^2 + f + 1$ colors are needed in the first coordinate. Thus it suffices to show that the number of colors used in the second coordinate is bounded by a function of d, e, T, and $\omega(G)$. In fact, we have already done so when we argued for Proposition 5.1, because the bound in the second coordinate of that coloring could be realized by applying First-Fit. □

6. The main theorem. In this section, we prove the following technical reformulation of Theorem 1.5, our central result.

THEOREM 6.1. *For every positive integer k, there exists an on-line coloring algorithm A_k and a function c_k such that, if $G^<$ is an on-line presentation of $G \in \mathrm{Forb}(T_k)$, then A_k gives $G^<$ a proper coloring using at most $c_k(\omega(G))$ colors.*

Proof. Let k be a fixed positive integer. We have already observed that it suffices to prove the theorem for $q\,\mathrm{Forb}(T_k)$. Another simplifying observation is that it suffices to prove the following, apparently weaker, statement:

(*) For every positive integer t, there exists an on-line algorithm $A_{k,t}$ and an absolute constant $c_{k,t}$ such that, if $G \in q\,\mathrm{Forb}(T_k)$ and $\omega(G) \le t$, then $A_{k,t}$ colors G using at most $c_{k,t}$ colors.

Statement (*) is, in fact, no weaker than Theorem 6.1; we simply use pairwise disjoint sets of colors for each algorithm $A_{k,t}$. If $G^<$ is an on-line presentation of a graph $G \in q\,\mathrm{Forb}(T_k)$, then, whenever $\omega(G_i^<) = \omega(G_{i-1}^<) = t$, we may color x_i using $A_{k,t}$. On the other hand, if $\omega(G_i^<) = \omega(G_{i-1}^<) + 1$, we may begin using $A_{k,t+1}$ and a new set of colors. We then have Theorem 6.1 by taking $c_k(\omega(G)) = \sum_{t=1}^{\omega} c_{k,t}$.

We now prove (*) by induction on the clique size t. If $t = 1$, the statement is obvious, since First-Fit will assign the same color to every vertex of a graph with no edges. Assume that $t > 1$ and that there exists an on-line algorithm $A_{k,t-1}$ and a constant $c_{k,t-1}$ such that $A_{k,t-1}$ colors any on-line presentation $G^<$ of $G \in q\,\mathrm{Forb}(T_k)$ satisfying $\omega(G) \le t - 1$, with at most $c_{k,t-1}$ colors. To prove the induction step, we must show that there exists an algorithm $A_{k,t}$ and a constant $c_{k,t}$ such that $A_{k,t}$ colors any on-line presentation $G^<$ of $G \in q\,\mathrm{Forb}(T_k)$ satisfying $\omega(G) \le t$, with at most $c_{k,t}$ colors; to this end, we set up a secondary induction. To state the secondary induction, we must refer to three sequences of parameters $p_2, \dots, p_t$, $\rho_2, \dots, \rho_t$, and $w_2, \dots, w_t$. We delay the calculation of these sequences until after a sketch of the secondary induction.

We call a template with s parts and $p_s + k^4$ vertices in each part an s-template. To appreciate the following statement, which we will prove by induction on s, it is important to realize that $p_s + k^4$ will be much smaller than w_s.

(**) For $2 \le s \le t + 1$, there exists an algorithm $A_{k,t,s}$ and a constant $c_{k,t,s}$ such that

(i) For $2 \le s \le t$, if $G^<$ is an on-line presentation of $G \in q\,\mathrm{Forb}(T_k, K_{w_s}^s)$ with $\omega(G) \le t$, then $A_{k,t,s}$ colors $G^<$ using at most $c_{k,t,s}$ colors,

(ii) For $3 \le s \le t + 1$, if $G^<$ is an on-line presentation of a graph $G \in q\,\mathrm{Forb}(T_k)$, where $\omega(G) \le t$ and no $(s-1)$-template of G has ρ_{s-1} extra points, then $A_{k,t,s}$ colors $G^<$ with at most $c_{k,t,s}$ colors.

Some general comments are in order now. The first comment is that the base step, the case where $s = 2$, follows from Theorem 1.4, regardless of the value of w_2. Note that (ii) makes no assertion in this case. By far, the hardest part of proving (**) is showing that, if (i) is true for $s - 1$, then (ii) is true for s. Next, the sequences of parameters will be defined in such a way that, whenever we have (ii) for a particular value of s, $3 \le s \le t$, we obtain (i) for the same value of s as an immediate corollary. Finally, we will define $\rho_t = 1$, so that (ii) in the case where $s = t + 1$ implies that we may prove the primary induction by putting $A_{k,t} = A_{k,t,t+1}$ and $c_{k,t} = c_{k,t,t+1}$. If $\omega(G) \le t$, a t-template of G cannot have any extra points, since no vertex of G can have neighbors in every part of a t-template.

We now state the properties that our sequences of parameters must have for us to prove (**).

(a′) If $G \in q\,\mathrm{Forb}(K_{w_s}^s)$ and B is an $(s-1)$-template of G, then B has less than ρ_{s-1} extra points, for $3 \le s \le t$.

(b′) If G is a graph and B is an s-template of G that has k extra points, $x_1, \dots, x_k$, then $N(x_1) \cap \cdots \cap N(x_k) \cap B \ne \varnothing$, for $2 \le s \le t$.

Property (a′) is all we need to show that establishing (ii) for a particular value of s yields a proof of (i) for that same value. Suppose that we have defined our parameters so that (a′) holds and assume (ii) for some $s \leq t$. Let $G^<$ be an on-line presentation of $G \in q\ \mathrm{Forb}(T_k,\ K^s_{w_s})$ with $\omega(G) \leq t$. Assuming (ii), we know that, if $A_{k,t,s}$ uses more than $c_{k,t,s}$ colors on $G^<$, then some $(s-1)$-template of G has ρ_{s-1} extra points. However, by (a′), $G \notin q\ \mathrm{Forb}(K^s_{w_s})$, a contradiction. Property (b′) is used to handle a small technicality that arises when we show that, if (i) is true for $s-1$, then (ii) is true for s. A more detailed motivation for (b′) outside the context of the algorithm is impractical.

We now define our parameters, using a "reversed" induction. Let the function w be defined by $w(k, p, \rho) = (p + k^4)(1 + f_1(0, k, \rho) + f_1(1, k, \rho))$ for all positive integers k, p, and ρ.

Base: Let $p_t = k^5$. Let $\rho_t = 1$. Let $w_t = w(k, p_t, \rho_t)$.

Induction: Suppose that p_s, ρ_s, and w_s have been defined for $s > 2$. Then $p_{s-1} = \max\{k^4 w_s, k^5\}$, $\rho_{s-1} = w_s$, and $w_{s-1} = w(k, p_{s-1}, \rho_{s-1})$.

As is the case for property (b′), the definition of the function w is motivated by technicalities that arise in a detailed discussion of the algorithm.

We now verify (a′). Suppose that $G \in q\ \mathrm{Forb}(K^s_{w_s})$ and B is a template of G with $s-1$ parts, $p_{s-1} + k^4$ vertices in each part, where $3 \leq s \leq t$. Suppose for contradiction that B has $r = \rho_{s-1} = w_s$ extra points, $x_1, \ldots, x_r$. Consider any fixed part Q of B. By the definition of an extra point, x_1 has $p_{s-1} \geq k^4 w_s$ neighbors in Q. At least $k^4 w_s - k^4$ of these neighbors are neighbors of x_2, again by the definition of an extra point. At least $k^4 w_s - 2k^4$ of these points are neighbors of x_3. Continuing in this manner, we may find a subset of Q of size at least $k^4 w_s - ((\rho_{s-1} - 1)k^4) = w_s$, which is adjacent to all of $\{x_1, \ldots, x_r\}$. Since Q was chosen arbitrarily, we may do the same in every part of B. This results in an augmented $K^s_{w_s}$, a contradiction.

Property (b′) is verified in a similar manner. Suppose that G is a graph, that B is a template of G with s parts, $p_s + k^4$ vertices in each part, and that B has k extra points, $x_1, \ldots, x_k$. Then x_1 has at least $p_s \geq k^5 = kk^4$ neighbors in every part of B, at least $kk^4 - k^4$ of which are also neighbors of x_2, and so on. We then find a common neighbor for $x_1, \ldots, x_k$. (In fact, we find at least $kk^4 - (k-1)k^4 = k^4$ neighbors in each part of B.)

By our earlier remarks, to prove the induction step of (*), it suffices to prove (**). We have already noted that, in the base step of (**), (i) is a consequence of Theorem 1.4 and (ii) is trivial. By our remarks on property (a′), to show the induction step of (**), it suffices to show that, whenever (i) holds for $s-1$, (ii) holds for s, $2 < s \leq t+1$. By the primary induction hypothesis, there exists an algorithm $A_{k,t-1}$ and a constant $c_{k,t-1}$ such that, if $G^<$ is an on-line presentation of a graph $G \in q\ \mathrm{Forb}(T_k)$ with $\omega(G) \leq t-1$, then $A_{k,t-1}$ colors $G^<$ with at most $c_{k,t-1}$ colors. By the secondary induction hypothesis, there exists an algorithm $A_{k,t,s-1}$ and a constant $c_{k,t,s-1}$ such that, if $G^<$ is an on-line presentation of $G \in q\ \mathrm{Forb}(T_k,\ K^{s-1}_{w_{s-1}})$ with $\omega(G) \leq t$, then $A_{k,t,s-1}$ colors G with at most $c_{k,t,s-1}$ colors. It remains for us to show that, given these hypotheses, there exists an on-line algorithm $A_{k,t,s}$ and a constant $c_{k,t,s}$ satisfying (ii).

For the remainder of the proof, let $p = p_{s-1}$ and $\rho = \rho_{s-1}$. Also, "template" should be read as "$(s-1)$-template." Recall that, if B is a template, then we call a vertex x a strong 1-neighbor of B if x has k neighbors in some part of B. We call x an extra point of B if x has less than k^4 nonneighbors in every part of B. An acceptable template sequence is a template sequence $B_1, \ldots, B_r$ such that, if $1 \leq m < n \leq r$ and $x \in B_n$, then x is not a strong 1-neighbor of B_m.

We now present the algorithm $A_{k,t,s}$. To show (ii), we assume that no template of the graph being presented has ρ_{s-1} extra points. The key feature of the algorithm is that it maintains an acceptable template sequence $B_1, \ldots, B_r$. Whenever a template B_i is added to the sequence, we arbitrarily assign labels $\{1, \ldots, |B_i|\}$ to the vertices of B_i. These labels are not part of the coloring, since a vertex may not become part of a template until long after it has entered the graph. To each template B_i, we will also associate a set of (not necessarily all) strong 1-neighbors, O_i. A vertex x may be assigned to O_i in one of two ways, either when x enters the graph (if B_i had already been formed) or when B_i is formed (if the template doesn't appear until after x has entered). In the latter case, we give x a "shadow" color, as we detail below. As with the labels on the template points, the shadow colors are not part of the coloring produced by the algorithm, but are instead records to be used internally by the algorithm. In either case, the assignment of x to O_i is irrevocable, and the O_i's are pairwise disjoint.

Suppose that when a vertex x enters the graph the acceptable sequence of templates is $B_1, \ldots, B_r$. The algorithm colors x and updates the template list as follows.

Case 1. If x is a strong 1-neighbor of some template in the sequence, find the smallest i such that x is a strong 1-neighbor of B_i. Add x to O_i. Assign x a several coordinate color. The first coordinate identifies x as a vertex that was classified as a strong 1-neighbor at the time it entered the graph. The second coordinate is the set of labels used on $N(x) \cap B_i$. Note that there are a fixed number of labels since all the templates have the same size. To compute the third coordinate, apply the algorithm $A_{k,t-1}$ (which exists by the primary induction hypothesis) to the subgraph induced by vertices of O_i that received the same colors as x in their first two coordinates. Since these vertices have a common neighbor in B_i, the induction hypothesis implies that we use at most $c_{k,t-1}$ colors in this coordinate. Let V' be the set of vertices that received the same color as x in their first three coordinates. Note that $V' \cap O_j$ is an independent set for $j = 1, \ldots, r$.

Claim A. The subgraph induced by V' and the independent sets $V' \cap O_j$ satisfy conditions (a) and (b) of Lemma 5.2 with $d = (k + \rho)f_1(1, k, \rho)$, $e = k + \rho$, and $f = (s - 1)(p + k^4)f_1(1, k, \rho)f_1(2, k, \rho)$.

Given this claim, by Lemma 5.2, we may apply an on-line algorithm to the subgraph induced by vertices that received the same color as x in their first three coordinates. The number of colors used in each coordinate will be bounded in terms of k, ρ, s, and $\omega(G)$, all of which are bounded in terms of t, and it will be a proper coloring.

Case 2. If x is not a strong 1-neighbor for any template in the sequence at the time x enters, then we attempt to find a set of vertices that, together with x, form a template that may be added to the sequence without violating the key properties of the sequence. That is, we look for a set B_{r+1} such that B_{r+1} is an $(s - 1)$-template and $B_{r+1} \cap (B_i \cup O_i) = \varnothing$ for $1 \le i \le r$. Note that, if B_{r+1} satisfies this condition, no vertex of B_{r+1} is a strong 1-neighbor of any earlier template in the sequence. If such a set B_{r+1} can be found, add B_{r+1} to the template sequence and assign labels to the vertices of B_{r+1}. Assign x a two-coordinate color. The first coordinate identifies x as a vertex that was used to form a new template at the time it entered. The second coordinate is computed by applying First-Fit to the subgraph induced by vertices that received the same color as x in their first coordinate. Note that every template in the acceptable sequence contains precisely one such vertex. Since we use First-Fit in the second coordinate, we know that x will be properly colored. Moreover, it is easy to see that the algorithm will use a bounded number of colors in the second coordinate for the subgraph induced by vertices that were used to form templates at the time they entered. This is because, by Lemma 4.1 and the fact that x is not an extra point of any previous template (it is not even a strong 1-neighbor), this subgraph has degree at most $f_1(0, k, \rho)$.

If y is a vertex that entered before x such that, for $1 \le i < r$, $y \notin B_i \cup O_i$ (with y a strong 1-neighbor of B_{r+1}), then assign y to O_{r+1}. The algorithm then assigns to y a "shadow" color $c(y)$; this color is strictly for record-keeping purposes, since y's "real" color was assigned when y entered. The shadow color is assigned as follows. Imagine that a "twin" vertex y' is presented immediately after x and that y' has precisely the same neighbors as y (at the time x enters and thereafter). Apply the algorithm to y' as if it were an actual point presented and, for any points in the graph that have already received a shadow color, use the shadow color, rather than the color actually assigned, to compute the color for y'. In fact, y' would be colored under Case 1, because y' looks exactly like y, which is now a strong 1-neighbor of B_{r+1}. When other vertices require shadow colors in the future, y' will be treated as if it had been actually presented. This will guarantee that the shadow colors form a proper coloring of the set of vertices that received shadow colors. If there is more than one such y, say $y_1, \ldots, y_m$, when B_{r+1} is formed, then apply the same procedure to each vertex. Finally, the algorithm assigns shadow colors to all the vertices of B_{r+1}, except x.

Claim B. If x is a vertex with shadow color $c(x)$, then $c(x) \notin S(x) = \{c(y): y \in N(x)\}$.

Case 3. If x is not a strong 1-neighbor of any template at the time x entered and if x cannot be used to form a new template for the sequence, then we assign x a three-coordinate color as follows. The first coordinate identifies x as a vertex that could not be colored in either Case 1 or Case 2. The second coordinate of x is the set $S(x) = \{c(y): y \in N(x)\}$. Note that the number of colors used in the second coordinate is 2^b, where b is the maximum number of colors used in step 1; assuming Claim A, b is bounded in terms of t, so 2^b is, as well. To compute the third coordinate of x, apply the algorithm $A_{k,t,s-1}$ (whose existence is asserted by the secondary induction hypothesis) to the subgraph G_S induced by vertices that received the same colors as x in their first two coordinates, where $S = S(x)$ is the second coordinate of x's color.

Claim C. The subgraph G_S does not contain an augmented $K^{s-1}_{w_{s-1}}$.

Assuming Claim C, by the secondary induction hypothesis, $A_{k,t,s-1}$ gives a proper coloring of G_S, and therefore x is properly colored. Moreover, the secondary induction hypothesis implies that no more than $c_{k,t,s-1}$ colors are used in the third coordinate.

By the remarks included in the statement of the algorithm, to finish the proof of (**), it suffices to prove Claims A–C.

Proof of Claim A. Since it is the simpler of the two, we first verify that condition (b) of Lemma 5.2 is satisfied. That is, we check that, whenever V' contains an augmented $K_{d,d}$, say H, there exist integers α and β such that $H \cap (O_\alpha \cup O_\beta)$ contains an augmented $K_{e,e}$, where $d = (k + \rho)f_1(1, k, \rho)$ and $e = k + \rho$. Consider a complete bipartite graph H in V' with $(k + \rho)f_1(1, k, \rho)$ vertices in each part. Let H_1 and H_2 be the independent sets of H. If $y \in H_1$, then $y \in O_j$, for exactly one j, $1 \leq j \leq r$. Note that, if $I = \{j : \exists y \in O_j \cap H_1\}$, $|I| < f_1(1, k, \rho)$. To see this, suppose without loss of generality that $\{1, \ldots, f_1(1, k, \rho)\}$ is contained in I. Then, since no template has ρ extra points and $|H_1| > \rho f_1(1, k, \rho)$, by the Pigeonhole Principle, some vertex y' of H_2 is not an extra point of templates $B_1, \ldots, B_g$, where $g = f_1(1, k, \rho)$. Then, however, we have a contradiction of Lemma 4.1, since y' is adjacent to all of H_1. Because of this bound on $|I|$, using the Pigeonhole Principle, we may find an index α such that $|O_\alpha \cup H_1| = k + \rho$. By a similar argument, we find a subset of $H_2 \cap O_\beta$ of size $k + \rho$. This completes the verification of condition (b).

We now verify (a) of Lemma 5.2; that is, for all α, $1 \leq \alpha \leq r$, $|\{\beta: (V' \cap O_\alpha) \cup (V' \cap O_\beta)$ contains an augmented $K_{e,e}\}| \leq f$, where $e = k + \rho$ and $f = (s - 1) \times (p + k^4)f_1(1, k, \rho)f_1(2, k, \rho)$. Consider an arbitrary node B_α of the auxiliary graph (or, more precisely, the node of the auxiliary graph corresponding to the template B_α). We wish to show first that, for every edge in the auxiliary graph that is incident on B_α (say the other endpoint is B_β), there exist a vertex $x_\beta \in B_\alpha$ and a strong 1-neighbor y_β of B_α such that

(i) y_β is at the top level of a 2-tube with 0-level B_β,

(ii) x_β and y_β are not extra points of B_β,

(iii) $x_\beta \sim y_\beta$.

This is illustrated in Fig. 6.1.

First, note that, if there exists a complete bipartite graph with $k + \rho$ vertices in each part where one part, say X, is contained in $V' \cap O_\alpha$ and the other, say Y, in $V' \cap O_\beta$ for some $\alpha \neq \beta$, then there exists a 2-tube such that its 0-level is B_β, its first level is contained in Y (choose any k vertices of Y), and the vertex y_β at the top level of the tube is an element of X that is not an extra point of B_β. Note that y_β exists because some vertex of X must fail to be an extra point of B_β, since $|X| > \rho$. Moreover, since every vertex of X is a strong 1-neighbor of B_α, we may find k vertices in one part of B_α that are neighbors of y_β. One of these k vertices must fail to be an extra point of B_α: if $\alpha > \beta$, then no element of B_α is an extra point (or even a strong 1-neighbor) of B_β. If $\beta > \alpha$ and k vertices in one part of B_α are extra points of B_β, then they have a common neighbor z in B_β, by property (b′). Then, however, z is a strong 1-neighbor of B_α, and this fact would have prevented us from adding B_α to the template sequence. Thus we may choose x_β to be any neighbor of y_β that is not an extra point of B_β.

Thus, if the degree of B_α is $(s-1)(p+k^4)f_1(1, k, \rho)f_1(2, k, \rho)$ or higher, we find, by the Pigeonhole Principle (there are only $(s-1)(p+k^4)$ vertices in B_α), a vertex x of B_α that has a neighbor in the top level of $f_1(1, k, \rho)f_1(2, k, \rho)$ 2-tubes with distinct 0-levels. (That is, $x = x_\beta$ for $f_1(1, k, \rho)f_1(2, k, \rho)$ distinct values of β.) See Fig. 6.2. Since the algorithm forces the O_i's to be pairwise disjoint, the 1-levels of these tubes are pairwise disjoint. It is still possible, however, that the 2-levels may not be disjoint. For each β, however, we choose the y_β so that it will not be extra point of B_β. Thus no vertex $y = y_\beta$ can be at the top of $f_1(1, k, \rho)$ of these tubes, or else there exist $f_1(1, k, \rho)$ disjoint 1-tubes whose bases form an acceptable template sequence and whose top levels are all adjacent to y, contradicting Lemma 4.1. Thus we may find among the collection of 2-tubes whose top points are adjacent to x a subcollection of $f_1(2, k, \rho)$ pairwise disjoint 2-tubes. When these tubes are ordered so that their bases are a subsequence of the ac-

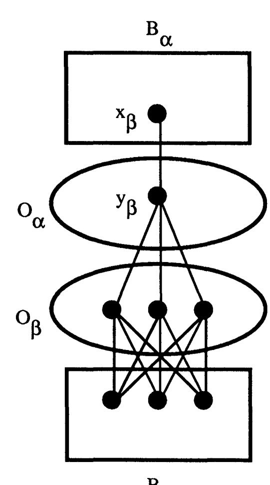

FIG. 6.1.

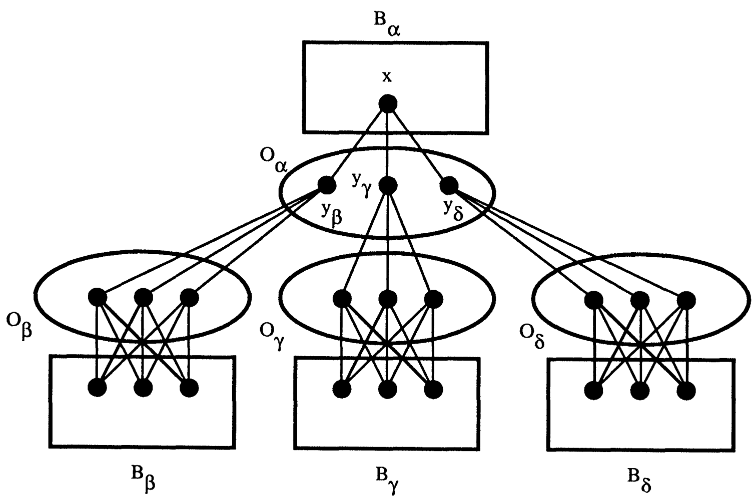

FIG. 6.2.

ceptable template sequence generated by the algorithm, the hypotheses of Lemma 4.1 are satisfied, so this is a contradiction. This shows that (a) of Lemma 5.2, i.e., the degree condition for the auxiliary graph, holds as claimed. □

Proof of Claim B. If $c(x)$ is the shadow color of x and $c(x) \in S(x)$, then there exists some vertex y with shadow color $c(y) = c(x)$ such that y received its shadow color before x entered the graph and $y \sim x$. However, since $y \sim x$ and y had a shadow color at the time the shadow color of x was assigned, $c(y) \neq c(x)$. □

Proof of Claim C. We argue by contradiction. Suppose that, at some point in the algorithm, some G_S contains an augmented $K^{s-1}_{w_{s-1}}$. Let Q be this subgraph and let $Q_1, \ldots, Q_{s-1}$ be the parts of Q. Let x be the last vertex of Q to enter the graph. Without loss of generality, $x \in Q_{s-1}$. Let $B_1, \ldots, B_r$ be the templates in the sequence at the time x entered.

We first claim that, for $1 \leq j \leq s-2$, no element of Q_j has been added to any B_j or O_j before x entered; if there were such a vertex, say y, then y would have received a shadow color $c(y)$. Then, however, $c(y) \in S(x) = S = S(y)$, contradicting Claim B.

Now consider Q_{s-1}. Let z be the last vertex of $Q' = Q_1 \cup \cdots \cup Q_{s-1}$ to enter the graph. Note that, by the above argument, z is not an extra point of any template in the sequence at the time x entered. (Indeed, this is true of all the vertices in Q'.) Label the vertices of Q_{s-1} as follows. For each $y \in Q_{s-1}$, assign y, if possible, the smallest m such that, at the time x entered, either $y \in B_m$ or $y \in O_m$. If no such m exists, assign y the label ∞. Note that the fact that z is not an extra point of any template in the sequence at the time x entered and the fact that z is adjacent to all of Q_{s-1} imply that at most $1 + f_1(0, k, \rho) + f_1(1, k, \rho)$ labels are used. Thus we find a set of vertices of size $p + k^4$ all of whose elements have the same label. In this case, however, ∞ cannot be the common label, or else the algorithm would have been able to add some template from Q to the sequence (and thereby give some vertex in Q a color according to Case 2). Thus the common label is a natural number m. Let $x_1, \ldots, x_a$, where $a = p + k^4$, be the vertices of Q_{s-1} that received the label m. Each of $x_1, \ldots, x_a$ must have entered the graph before B_m was formed; any that entered after B_m was formed would have been a strong 1-neighbor at the time it entered, and thus would have been colored in Case 1. On the other hand, z must have entered after B_m was formed: by the manner in which sets of strong 1-neighbors are formed, $x_1, \ldots, x_a$ are not strong 1-neighbors of any template preceding B_m in the sequence; else, they would have been assigned to, say, O_n where $n < m$, as soon as B_n was formed. Hence, if z, and therefore all of Q', entered before B_m was formed, the algorithm would have added a template to the sequence and the last of the points $z, x_1, \ldots, x_a$ to enter the graph would have been colored by Case 2. It cannot be the case that z entered at the time B_m was formed; else, z would have been colored in Case 2. If, however, z enters G after B_m is formed, then, x_1, for example, had received a shadow color by the time z entered, so $c(x_1) \in S(z)$. Then $c(x_1) \in S(z) - S(x_1)$ and $S(z) \neq S(x_1)$, a contradiction. $\square$

This completes the induction step of (**). The induction step of (*) follows, and the proof of the theorem is complete. $\square$

REFERENCES

[B] D. BEAN, *Effective coloration*, J. Symbolic Logic, 41 (1976), pp. 469–480.

[C] V. CHVÁTAL, *Perfectly ordered graphs*, in Topics on Perfect Graphs, Ann. Discrete Math., 21 (1989), pp. 63–65.

[EH] P. ERDÖS AND A. HAJNAL, *On chromatic number of graphs and set systems*, Acta Math. Acad. Sci. Hungar., 17 (1966), pp. 61–99.

[G1] A. GYÁRFÁS, *On Ramsey Covering-Numbers*, Coll. Math. Soc. János Bolyai 10, Infinite and Finite Sets, North–Holland/American Elsevier, New York, 1975, pp. 801–816.

[G2] ———, *Problems from the world surrounding perfect graphs*, Zastosowania Matematyki Applicationes Mathemacticae, 19 (1985), pp. 413–441.

[Ga] T. GALLAI, *Transitiv orientierbare Graphen*, Acta. Math. Acad. Sci. Hungar., 18 (1967), pp. 25–66.

[GL1] A. GYÁRFÁS AND J. LEHEL, *On-line and first-fit coloring of graphs*, J. Graph Theory, 12 (1988), pp. 217–227.

[GL2] ———, *First-fit and on-line chromatic number of families of graphs*, Ars Combinatoria, 29C (1990), pp. 168–176.

[GL3] ———, *Effective on-line coloring of P_5-free graphs*, Combinatorica, 11 (1991), pp. 181–184.

[GRS] R. GRAHAM, B. ROTHSCHILD, AND J. SPENCER, *Ramsey Theory*, John Wiley, New York, 1980.

[GST] A. GYÁRFÁS, E. SZEMERÉDI, AND TUZA, *Induced subtrees in graphs of large chromatic number*, Discrete Math., 30 (1980), pp. 235–244.

[K1] H. A. KIERSTEAD, *An effective version of Dilworth's theorem*, Trans. Amer. Math., Soc., 268 (1981), pp. 63–77.

[K2] ———, *Recursive colorings of highly recursive graphs*, Canad. J. Math., 33 (1981), pp. 1279–1290.

[K3] ———, *An effective version of Hall's theorem*, Proc. Amer. Math. Soc., 88 (1983), pp. 124–128.

[K4] ———, *The linearity of First-Fit for coloring interval graphs*, SIAM J. Discrete Math., 1 (1988), pp. 526–530.

[KP1] H. A. KIERSTEAD AND S. G. PENRICE, *Radius two trees specify χ-bounded classes*, J. Graph Theory, to appear.

[KP2] ———, *Recent results on a conjecture of Gyárfás*, Congr. Numer., 79 (1990), pp. 182–186.

[KPT] H. A. KIERSTEAD, S. G. PENRICE, AND W. T. TROTTER, *On-line and first-fit colorings of graphs which do not induce P_5*, SIAM J. Discrete Math., submitted.

[KR] H. A. KIERSTEAD AND V. RÖDL, *Applications of Hypergraph Coloring to Coloring Graphs Which Do Not Induce Certain Trees*, preprint.

[KT1] H. A. KIERSTEAD AND W. T. TROTTER, *An extremal problem in recursive combinatorics*, Congr. Numer., 33 (1981), pp. 143–153.

[KT2] ———, *Colorful induced subgraphs*, Discrete Math., 101 (1992), pp. 165–169.

[S1] J. H. SCHMERL, *Recursive colorings of graphs*, Canad. J. Math., 32 (1980), pp. 821–831.

[S2] ———, *The effective version of Brooks' theorem*, Canad. J. Math., 34 (1982), pp. 1036–1046.

[Su] D. P. SUMNER, *Subtrees of a graph and chromatic number*, in The Theory and Applications of Graphs, G. Chartrand, ed., John Wiley, New York, 1981, pp. 557–576.

[W] D. R. WOODALL, *Problem No. 4, Combinatorics*, in Proc. British Combinatorial Conference, 1973, London Math. Soc. Lecture Note Series 13, T. P. McDonough and V. C. Marvon, eds., Cambridge University Press, Cambridge, UK, 1974, p. 202.