The Mondshein Sequence*

Jens M. Schmidt

TU Ilmenau

Abstract. Canonical orderings [STOC'88, FOCS'92] have been used as a key tool in graph drawing, graph encoding and visibility representations for the last decades. We study a far-reaching generalization of canonical orderings to non-planar graphs that was published by Lee Mondshein in a PhD-thesis at M.I.T. as early as 1971.

Mondshein proposed to order the vertices of a graph in a sequence such that, for any i, the vertices from 1 to i induce essentially a 2-connected graph while the remaining vertices from $i+1$ to n induce a connected graph. Mondshein's sequence generalizes canonical orderings and became later and independently known under the name *non-separating ear decomposition*. Currently, the best known algorithm for computing this sequence achieves a running time of $O(nm)$; the main open problem in Mondshein's and follow-up work is to improve this running time to a subquadratic time.

In this paper, we present the first algorithm that computes a Mondshein sequence in time and space $O(m)$, improving the previous best running time by a factor of n. In addition, we illustrate the impact of this result by deducing linear-time algorithms for several other problems, for which the previous best running times have been quadratic.

In particular, we show how to compute three independent spanning trees in a 3-connected graph in linear time, improving a result of Cheriyan and Maheshwari [J. Algorithms 9(4)]. Secondly, we improve the preprocessing time for the output-sensitive data structure by Di Battista, Tamassia and Vismara [Algorithmica 23(4)] that reports three internally disjoint paths between any given vertex pair from $O(n^2)$ to $O(m)$. Thirdly, we improve the computation of 3-partitioning of a 3-connected graph to linear time. Finally, we show how a very simple linear-time planarity test can be derived once a Mondshein sequence is computed.

1 Introduction

Canonical orderings are a fundamental tool used in graph drawing, graph encoding and visibility representations; we refer to [1] for a wealth of applications. For maximal planar graphs, canonical orderings were first introduced by de Fraysseix, Pach and Pollack [6,7] in 1988. Kant then generalized canonical orderings to arbitrary 3-connected planar graphs [12,13].

Surprisingly, the concept of canonical orderings can be traced back much further, namely to a long-forgotten PhD-thesis at M.I.T. by Lee F. Mondshein [15]

* This research was partly done at Max Planck Institute for Informatics, Saarbrücken.

J. Esparza et al. (Eds.): ICALP 2014, Part I, LNCS 8572, pp. 967–978, 2014.

in 1971. In fact, Mondshein proposed a sequence that generalizes canonical orderings to non-planar graphs, hence making them applicable to arbitrary 3-connected graphs. Mondshein's sequence was, independently and in a different notation, found later by Cheriyan and Maheshwari [4] under the name *non-separating ear decompositions.*

Computationally, it is an intriguing question how fast a Mondshein sequence can be computed. Mondshein himself gave an involved algorithm with running time $O(m^2)$. Cheriyan showed that it is possible to achieve a running time of $O(nm)$ by using a theorem of Tutte that proves the existence of non-separating cycles in 3-connected graphs [20]. Both works (see [15, p 1.2] and [4, p. 532]) state as main open problem, whether it is possible to compute a Mondshein sequence in subquadratic time.

We present the first algorithm that computes a Mondshein sequence in time and space $O(m)$, hence solving the above 40-year-old problem. The interest in such a computational result stems from the fact that 3-connected graphs play a crucial role in algorithmic graph theory; we illustrate this in four direct applications by giving linear-time (and hence optimal) algorithms for several problems, for two of which the previous best running times have been quadratic.

In particular, we show how to compute three independent spanning trees in a 3-connected graph in linear time, improving a result of Cheriyan and Maheshwari [4]. Second, we improve the preprocessing time from $O(n^2)$ to $O(m)$ for a data structure by Di Battista, Tamassia and Vismara [8] that reports three internally disjoint paths in a 3-connected graph between any given vertex pair in time $O(\ell)$, where ℓ is the total length of these paths. Finally, we illustrate the usefulness of Mondshein's sequence by giving a very simple linear-time planarity test, once a Mondshein sequence is computed.

We start by giving an overview of Mondshein's work and its connection to canonical orderings and non-separating ear decompositions in Section 3. Section 4 sketches the main ideas for our linear-time algorithm that computes a Mondshein sequence. Section 5 covers four applications of this linear-time algorithm.

2 Preliminaries

We use standard graph-theoretic terminology and assume that all graphs are simple.

Definition 1 ([14,23])**.** An *ear decomposition* of a 2-connected graph $G = (V, E)$ is a sequence $(P_0, P_1, \ldots, P_k)$ of subgraphs of G that partition E such that P_0 is a cycle and every P_i, $1 \leq i \leq k$, is a path that intersects $P_0 \cup \cdots \cup P_{i-1}$ in exactly its end points. Each P_i is called an *ear*. An ear is *short* if it is an edge and *long* otherwise.

According to Whitney [23], every ear decomposition has exactly $m - n + 1$ ears. For any i, let $G_i = P_0 \cup \cdots \cup P_i$ and $\overline{V_i} := V - V(G_i)$. We write $\overline{G_i}$ to denote the graph induced by $\overline{V_i}$. We observe that $\overline{G_i}$ does not necessarily contain all edges in $E - E(G_i)$; in particular, there may be short ears in $E - E(G_i)$ that have both of their endpoints in G_i.

For a path P and two vertices x and y in P, let $P[x, y]$ be the subpath in P from x to y. A path with endpoints v and w is called a *vw-path.* A vertex x in a vw-path P is an *inner vertex* of P if $x \notin \{v, w\}$. For convenience, every vertex in a cycle is an inner vertex of that cycle.

The set of inner vertices of an ear P is denoted as $inner(P)$. The inner vertex sets of the ears in an ear decomposition of G play a special role, as they partition $V(G)$. Every vertex of G is contained in exactly one long ear as inner vertex. This gives readily the following characterization of $\overline{V_i}$.

Observation 2. *For every i, $\overline{V_i}$ is the union of the inner vertices of all long ears P_j with $j > i$.*

We will compare vertices and edges of G by their first occurrence in a fixed ear decomposition.

Definition 3. Let $D = (P_0, P_1, \ldots, P_{m-n})$ be an ear decomposition of G. For an edge $e \in G$, let $birth_D(e)$ be the index i such that P_i contains e. For a vertex $v \in G$, let $birth_D(v)$ be the minimal i such that P_i contains v (thus, $P_{birth_D(v)}$ is the ear containing v as an inner vertex). Whenever D is clear from the context, we will omit D.

Clearly, for every vertex v, the ear $P_{birth(v)}$ is long, as it contains v as an inner vertex.

3 Generalizing Canonical Orderings

We give a compact rephrasing of canonical orderings in terms of non-separating ear decompositions. This will allow for an easier comparison of a canonical ordering and its generalization to non-planar graphs, as the latter is also based on ear decompositions. We assume that the input graphs are 3-connected and, when talking about canonical orderings, planar. It is well-known that maximal planar graphs, which were considered in [6], form a subclass of 3-connected graphs (apart from the triangle-graph).

Definition 4. An ear decomposition is *non-separating* if, for $0 \leq i \leq m - n$, every inner vertex of P_i has a neighbor in $\overline{G_i}$ unless $\overline{G_i} = \emptyset$.

The name *non-separating* refers to the following helpful property.

Lemma 5. *In a non-separating ear decomposition D, $\overline{G_i}$ is connected for every i.*

Proof. Let u be an inner vertex of the last long ear in D. If $\overline{G_i} = \emptyset$, the claim is true. Otherwise, consider any vertex x in $\overline{G_i}$. In order to show connectedness, we exhibit a path from x to u in $\overline{G_i}$. If x is an inner vertex of $P_{birth(u)}$, this path is just the path $P_{birth(u)}[x, u]$. Otherwise, $birth(x) < birth(u)$. Then x has a neighbor in $\overline{G_{birth(x)}}$, since D is non-separating, and, according to Observation 2, this neighbor is an inner vertex of some ear P_j with $j > birth(x)$. Applying induction on j gives the desired path to u. □

A *plane graph* is a graph that is embedded into the plane. In particular, a plane graph has a fixed outer face. We define canonical orderings as special non-separating ear decompositions.

Definition 6 (canonical ordering)**.** Let G be a 3-connected plane graph having the edges tr and ru in its outer face. A *canonical ordering* with respect to tr and ru is an ear decomposition D of G such that

1. $tr \in P_0$,
2. $P_{birth(u)}$ is the last long ear, contains u as its only inner vertex and does not contain ru, and
3. D is non-separating.

The original definition of canonical orderings by Kant [13] states several additional properties, all of which can be deduced from the ones given in Definition 6. E.g., it is easy to see for every i that the outer face C_i of G_i forms a cycle containing tr.

The fact that D is non-separating plays a key role for both canonical orderings and their generalization to non-planar graphs. E.g., for canonical orderings, Lemma 5 implies that the plane graph G can be constructed from P_0 by successively inserting the ears of D to only one dedicated face of the current embedding, a routine that is heavily applied in graph drawing and embedding problems.

Our definition of canonical orderings uses planarity only in one place: $tr \cup ru$ is assumed to be part of the outer face of G. Note that the essential art of this assumption is that $tr \cup ru$ is part of some face of G, as we can always choose an embedding for G having this face as outer face. By dropping this assumption, our definition of canonical orderings can be readily generalized to non-planar graphs: We merely require tr and ru to be edges in the graph.

This is in fact equivalent to the definition Mondshein used 1971 to define a *(2,1)-sequence* [15, Def. 2.2.1], but which he gave in the notation of a special vertex ordering. This vertex ordering actually refines the partial order $inner(P_0), \dots, inner(P_{m-n})$ by enforcing an order on the inner vertices of each path according to their occurrence on that path (in any direction). For conciseness, we will instead stick to the following short ear-based definition, which is similar to the one given in [4] but does not need additional degree-constraints.

Definition 7 ([15,4])**.** Let G be a graph with an edge ru. A *Mondshein sequence avoiding* ru (see Figure 3a) is an ear decomposition D of G such that

1. $r \in P_0$,
2. $P_{birth(u)}$ is the last long ear, contains u as its only inner vertex and does not contain ru, and
3. D is non-separating.

An ear decomposition D that satisfies Conditions 1 and 2 is said to *avoid* ru. Put simply, this forces ru to be "added last" in D, i.e., strictly after the last long ear $P_{birth(u)}$ has been added. Note that Definition 7 implies $u \notin P_0$, as $P_{birth(u)}$ contains only one inner vertex. As a direct consequence of this and the fact that D is non-separating, G must have minimum degree 3 in order to have a Mondshein sequence. Mondshein proved that every 3-connected graph has a Mondshein sequence. In fact, also the converse is true.

Theorem 8. *[4,24] Let $ru \in E(G)$. Then G is 3-connected if and only if G has a Mondshein sequence avoiding ru.*

We state two additional facts about Mondshein sequences. Since we replaced the assumption that $tr \cup ru$ is in the outer face of G with the very small assumption that ru is an edge of G (which does not assume anything about t at all), it is natural to ask how we can extract t (and thus, a canonical ordering) from a Mondshein sequence when G is plane. We choose t as any neighbor of r in P_0. Since P_0 is non-separating and the non-separating cycles of a 3-connected plane graph are precisely its faces [20], this satisfies Definition 6 and leads to the following observation.

Observation 9. *Let D be a Mondshein sequence avoiding ru of a planar graph G and let t be a neighbor of r in P_0. Then D is a canonical ordering of the planar embedding of G whose outer face contains $tr \cup ru$.*

Once having a Mondshein sequence, we can aim for a slightly stronger structure. A *chord* of an ear P_i is an edge in G that joins two non-adjacent vertices of P_i. Let a Mondshein sequence be *induced* if P_0 is induced in G and every ear $P_i \neq P_0$ has no chord in G, except possibly the chord joining the endpoints of P_i. The following lemma shows that we can always expect Mondshein sequences that are induced. We omit the proof.

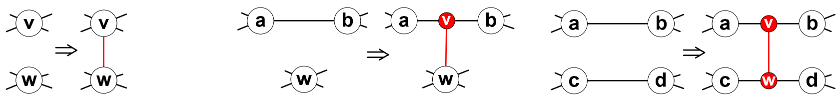
Lemma 10. *Every Mondshein sequence can be transformed to an induced Mondshein sequence in linear time.*

4 Computing a Mondshein Sequence

Mondshein gave an involved algorithm [15] that computes his sequence in time $O(m^2)$. Independently, Cheriyan and Maheshwari gave an algorithm that runs in time $O(nm)$ and which is based on a theorem of Tutte. At the heart of our linear-time algorithm is the following classical construction sequence for 3-connected graphs due to Barnette and Grünbaum [2] and Tutte [21, Thms. 12.64 and 12.65].

Definition 11. The following operations on simple graphs are *BG-operations* (see Figure 1).

(a) *vertex-vertex-addition*: Add an edge between two distinct non-adjacent vertices
(b) *edge-vertex-addition*: Subdivide an edge ab, $a \neq b$, by a vertex v and add the edge vw for a vertex $w \notin \{a, b\}$
(c) *edge-edge-addition*: Subdivide two distinct edges by vertices v and w, respectively, and add the edge vw



(a) vertex-vertex-addition (b) edge-vertex-addition (c) edge-edge-addition

Fig. 1. BG-operations

Theorem 12 ([2,21])**.** *A graph is 3-connected if and only if it can be constructed from K_4 using BG-operations.*

Hence, applying an BG-operation on a 3-connected graphs preserves it to be simple and 3-connected. Let a *BG-sequence* of a 3-connected graph G be a sequence of BG-operations that constructs G from K_4. It has been shown that such a BG-sequence can be computed efficiently.

Theorem 13 ([17, Thms. 6.(2) and 52])**.** *A BG-sequence of a* 3*-connected graph can be computed in time $O(m)$.*

The outline of our algorithm is as follows. We start with a Mondshein sequence of K_4, which is easily obtained, and compute a BG-sequence of our 3-connected input graph by using Theorem 13. The crucial part is now a careful analysis that a Mondshein sequence of a 3-connected graph G can be modified to one of G', where G' is obtained from G by applying a BG-operation.

This last step is the main technical contribution of this paper and depends on the various positions in the sequence in which the vertices and edges that are involved in the BG-operation can occur. We will prove that there is always a modification that is local in the sense that the only long ears that are modified are the ones containing a vertex that is involved in the BG-operation.

Lemma 14 (Path Replacement Lemma)**.** *Let G be a* 3*-connected graph with an edge ru. Let $D = (P_0, P_1, \ldots, P_{m-n})$ be a Mondshein sequence avoiding ru of G. Let G' be obtained from G by applying a single BG-operation Γ and let ru' be the edge of G' corresponding to ru. Then a Mondshein sequence D' of G' avoiding ru' can be computed from D using only constantly many constant-time modifications.*

However, the complete description of these modifications goes beyond the scope of this extended abstract. We will therefore state precise modifications only for the very first cases of vertex-vertex- and edge-vertex-additions and omit everything else.

We need some notation for describing the modifications. Let vw be the edge that was added by Γ such that, if applicable, v subdivides $ab \in E(G)$ and w subdivides $cd \in E(G)$. Then the edge ru' of G' that *corresponds* to ru in G is either ru, rv or rw. Whenever we consider the edge ab or cd, e.g. in a statement about $birth(ab)$, we assume that Γ subdivides ab, respectively, cd. W.l.o.g., we further assume that $birth(a) \leq birth(b)$, $birth(c) \leq birth(d)$ and

$birth(d) \leq birth(b)$. If not stated otherwise, the *birth*-operator refers always to D in this section. Let $S \subseteq \{av, vb, vw, cw, wd\}$ be the set of new edges in G'.

We state the detailed replacement scheme that plays a key-role in proving the above Path Replacement Lemma.

Lemma 15. *There is a Mondshein sequence* $D' = (P'_0, P'_1, \ldots, P'_{m-n+1})$ *of* G' *avoiding* ru *(respectively,* rv *or* rw *if* Γ *subdivides* ru*) that can be obtained from* D *by performing the following four modifications:*

M1) replacing the long ear $P_{birth(b)}$ *with* $1 \leq i \leq 3$ *consecutive long ears* P'_{b_1}*,* P'_{b_2} *and* P'_{b_3}*, each of which consists of edges in* $P_{birth(b)} \cup S$ *(for notational convenience, we assume that all three ears exist such that* $P'_{b_j} := P'_{b_i}$ *for every* $j > i$*)*

M2) if $P_{birth(cd)}$ *is long and* $birth(d) < birth(b)$*, replacing* $P_{birth(cd)}$ *with the long ear* P'_{cwd} *that is obtained from* $P_{birth(cd)}$ *by subdividing* cd *with* w *(in particular,* $birth(cd) = birth(d) < birth(b)$ *in this case)*

M3) if $P_{birth(ab)}$ *is short, deleting or replacing* $P_{birth(ab)}$ *with an edge in* $\{va, vb, vw\}$*; if* $P_{birth(cd)}$ *is short, deleting or replacing* $P_{birth(cd)}$ *with an edge in* $\{wc, wd\}$

M4) possibly adding vw *as new last ear.*

In particular, D' *can be constructed from* D *as follows (Figure 2 determines the new ears* P'_{b_1}*–*P'_{b_3} *in* $M1$*).*

(1) Γ *is a vertex-vertex-addition*
  Obtain D' from D by adding the new ear vw at the end.

(2) Γ *is an edge-vertex-addition*
  (a) $birth(b) = birth(ab)$
    Let a' and b' be the endpoints of $P_{birth(b)}$ such that a' is closer to a than to b on $P_{birth(b)}$ (a' may be a, but $b' \neq b$).
    (i) $w \notin G_{birth(b)}$ $\triangleright$ $birth(w) > birth(b)$
      Obtain D' from D by subdividing $ab \subseteq P_{birth(b)}$ with v and adding the new ear vw at the end.
    (ii) $w \in G_{birth(b)} - P_{birth(b)}$ $\triangleright$ $birth(w) < birth(b)$ *and* $w \notin \{a', b'\}$
      Let Z be the path obtained from $P_{birth(b)}$ by replacing ab with $av \cup vb$. Let Z_1 be the $a'w$-path in $Z \cup vw$. Obtain D' from D by replacing $P_{birth(b)}$ with the two ears Z_1 and $Z[v, b']$ in that order.
    (iii) $w \in P_{birth(b)}$ $\triangleright$ $birth(w) = birth(b)$ *or* $w \in \{a', b'\}$
      Let Z be obtained from $P_{birth(b)}$ by replacing ab with $av \cup vb$. Let Z_2 be the vw-path in Z (if $birth(b) = 0$, Z is a cycle and there are two vw-paths; we then choose one that does not contain r as an inner vertex). Let Z_1 be obtained from Z by replacing Z_2 with the edge vw. Obtain D' from D by replacing $P_{birth(b)}$ with the two ears Z_1 and Z_2 in that order.

We omit a concise proof of the correctness of Lemma 15 and, thus, of the Path Replacement Lemma 14. Applying Lemma 14 iteratively for each operation in a BG-sequence gives immediately a linear-time algorithm for constructing

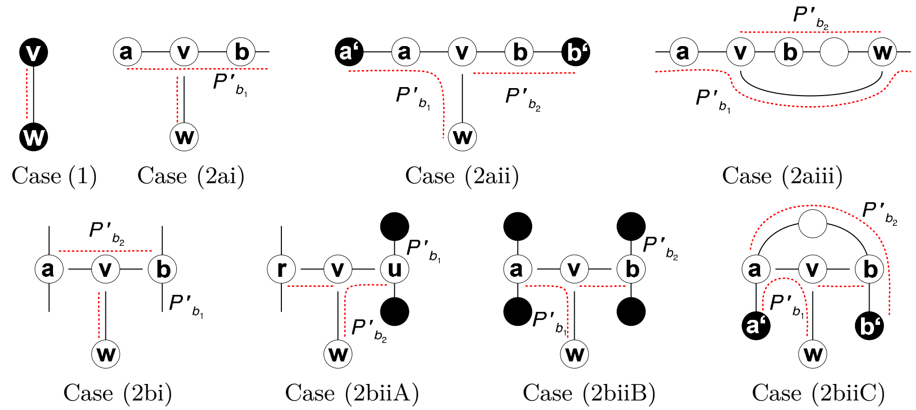


Fig. 2. Cases (1) and first subcases of (2) of Lemma 15. Black vertices are endpoints of ears that are contained in $G_{birth(b)}$. The dashed paths depict (parts of) the ears in D'.

a Mondshein sequence, as each step can be computed in constant time. We conclude the following theorem.

Theorem 16. *Given an edge ru of a 3-connected graph G, a Mondshein sequence of G avoiding ru can be computed in time $O(m)$.*

We now discuss four applications where Theorem 16 leads immediately to linear-time solutions. For three of these problems only quadratic algorithms have been known.

5 Applications

Application 1: Independent Spanning Trees
Let k spanning trees of a graph be *independent* if they all have the same root vertex r and, for every vertex $x \neq r$, the paths from r to x in the k spanning trees are *internally disjoint* (i.e., vertex-disjoint except for their endpoints). The following conjecture from 1988 due to Itai and Rodeh [11] has received considerable attention in graph theory throughout the past decades.

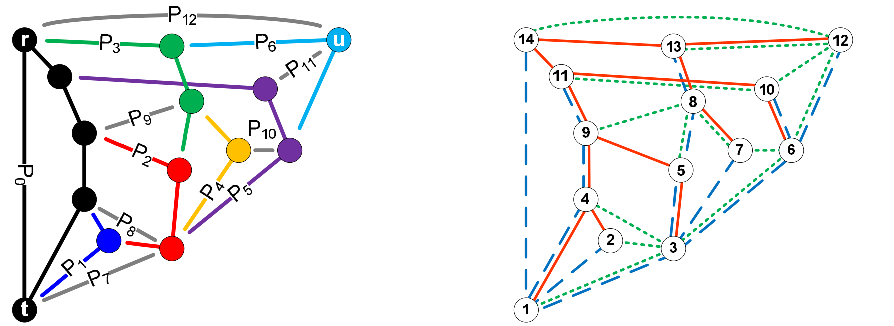
Conjecture (Independent Spanning Tree Conjecture [11])**.** Every k-connected graph contains k independent spanning trees.

The conjecture has been proven for $k \leq 2$ [11], $k = 3$ [4,24] and $k = 4$ [5], with running times $O(m)$, $O(n^2)$ and $O(n^3)$, respectively, for computing the corresponding independent spanning trees. For $k \geq 5$, the conjecture is open. For planar graphs, the conjecture has been proven by Huck [10].

We show how to compute three independent spanning trees in linear time, using an idea of [4]. This improves the previous best running time by a factor of n. It may seem tempting to compute the spanning trees directly and without using a Mondshein sequence, e.g. by local replacements in an induction over BG-operations or inverse contractions. However, without additional structure it can be proven that this is bound to fail.

Compute a Mondshein sequence avoiding ru, as described in Theorem 16. Choose r as the common root vertex of the three spanning trees and let $x \neq r$ be an arbitrary vertex.

First, we show how to obtain two internally disjoint paths from x to r that are both contained in the subgraph $G_{birth(x)}$. An *st-numbering* π is an ordering $v_1 < \cdots < v_n$ of the vertices of a graph such that $s = v_1$, $t = v_n$, and every other vertex has both a higher-numbered and a lower-numbered neighbor. Let π be *consistent* to a Mondshein sequence if π is an st-numbering for every graph G_i, $0 \leq i \leq m - n$. Let $t \neq u$ be a neighbor of r in P_0. A consistent tr-numbering π can be easily computed in linear time [3]. According to π, we can start with x and iteratively traverse to a higher-numbered and lower-numbered neighbor, respectively, without leaving $G_{birth(x)}$. This gives two internally disjoint paths from x to r and t; the path to t is then extended to the desired path ending at r by appending the edge tr. The traversed edges of this procedure for every $x \neq r$ give the first two independent spanning trees T_1 and T_2.



(a) A Mondshein sequence of a non-planar 3-connected graph G.

(b) Three independent spanning trees in G (vertex numbers depict a consistent st-numbering).

Fig. 3.

We construct the third independent spanning tree. Since a Mondshein sequence is non-separating, we can start with any vertex $x \neq r$, traverse to a neighbor in $\overline{G_{birth(x)}}$ and iterate this procedure until we end at u. The traversed edges of this procedure for every $x \neq r$ form a tree that is rooted at u and that can be extended to a spanning tree T_3 that is rooted at r by adding the edge ur. T_3 is independent from T_1 and T_2, as, for every $x \neq r$, the path from x to u intersects $G_{birth(x)}$ only in x.

Application 2: Output-Sensitive Reporting of Disjoint Paths
Given two vertices x and y of an arbitrary graph, a *k-path query* reports k internally disjoint paths between x and y or outputs that these do not exist. Di Battista, Tamassia and Vismara [8] give data structures that answer k-path queries for $k \leq 3$. A key feature of these data structures is that every k-path query has an *output-sensitive* running time, i.e., a running time of $O(\ell)$ if the total length of the reported paths is ℓ (and running time $O(1)$ if the paths do not exist). The preprocessing time of these data structures is $O(m)$ for $k \leq 2$ and $O(n^2)$ for $k = 3$.

For $k = 3$, Di Battista et al. show how the input graph can be restricted to be 3-connected using a standard decomposition. For every 3-connected graph we can compute a Mondshein sequence, which allows us to compute three independent spanning trees T_1–T_3 in a linear preprocessing time, as shown in Application 1. If x or y is the root r of T_1–T_3, this gives a straight-forward output-sensitive data structure that answers 3-path queries: we just store T_1–T_3 and extract one path from each tree per query.

In order to extend these queries to k-path queries between arbitrary vertices x and y, [8] gives a case distinction that shows that the desired paths can be efficiently found in the union of the six paths in T_1–T_3 that join x with r and y with r. This case distinction can be used for the desired output-sensitive reporting in time $O(\ell)$ without changing the preprocessing. We conclude a linear preprocessing time for all k-path queries with $k \leq 3$.

Application 3: Planarity Testing
We give a conceptually very simple planarity test based on Mondshein's sequence for any 3-connected graph G in time $O(n)$.

The 3-connectivity requirement is not really crucial, as the planarity of G can be reduced to the planarity of all 3-connected components of G, which in turn are computed as a side-product for the BG-sequence in Theorem 13; alternatively, one can use standard algorithms [9,16] for reducing G to be 3-connected. We compute an induced Mondshein sequence D avoiding an arbitrary edge ru in time $O(n)$. Let t be a neighbor of r in P_0.

We start with a planar embedding M_0 of P_0 and assume with Observation 9 w.l.o.g. that the last vertex u will be embedded in the outer face. We will first ignore short ears. Step by step, we attempt to augment M_i with the next long ear P_j in D in order to construct a planar embedding M_j of G_j.

Once the current embedding M_i contains u, we have added all the vertices of G and are done. Otherwise, u is contained in $\overline{G_i}$, according to Definition 6.2. Then $\overline{G_i}$ contains a path from each inner vertex of P_j to u, according to Lemma 5. Since u is contained in the outer face of the final embedding, adding the long ear P_j to M_i can preserve planarity only when it is embedded into the outer face f of M_i. Thus, we only have to check that both endpoints of P_j are contained in f (this is easy to test by maintaining the vertices of the current outer face). If yes, we embed P_j into f. Otherwise, we output "not planar"; if desired, a Kuratowski-subdivision can then be extracted in linear time.

Until now we ignored short ears, but have already constructed a planar embedding M' of a spanning subgraph of G. In order to test whether the addition of the short ears to M' can make the embedding non-planar, we pass through the construction of M' once more, this time adding short ears. Whenever a long ear P_j is embedded, we test whether all short ears that join a vertex of $inner(P_j)$ with a vertex of G_{j-1} can be embedded while preserving a planar embedding. Note that if D is a canonical ordering of M, G_{j-1} must be 2-connected and the outer face of G_{j-1} must be a cycle, according to [19, Corollary 1.3]. The last fact allows for an easy test whether adding the short ears preserves a planar embedding.

Application 4 (Bonus Application): The *k-partitioning* problem
At the time of submission, the author was pointed to the following problem. Given vertices $v_1, \ldots, v_k$ of a graph G and natural numbers $n_1, \ldots, n_k$ with $n_1 + \cdots + n_k = n$, find a partition of V into sets $V_1, \ldots, V_k$ with $|V_i| = n_i$ for every i such that every set V_i induces a connected graph in G.

For general graphs, this problem is NP-hard even for $k = 2$. However, for 3-connected graphs, the 3-partitioning problem can be solved in linear time if the input graph is *planar*. As suggested in [22], this problem (as well as a related extension) can be solved directly, once a non-separating ear decomposition has been computed. For planar graphs, we can thus use the (well-established) canonical ordering instead, which simplifies previous algorithms considerably.

More importantly, the fastest algorithm for the 3-partitioning problem in arbitrary 3-connected graphs runs in time $O(n^2)$ [18]. Combining a Mondshein sequence through with a simple assignment of the vertices on ears to V_1, V_2 and V_3 (as shown in [22]) gives the first $O(m)$ algorithm for this problem.

Acknowledgments. I wish to thank Joseph Cheriyan for valuable hints and helpfully providing a scan of his PhD-thesis, the anonymous person who drew my attention to Lee F. Mondshein's work, and the anonymous reviewer that pointed me to the k-partitioning problem.

References

1. Badent, M., Brandes, U., Cornelsen, S.: More canonical ordering. J. Graph Algorithms Appl. 15(1), 97–126 (2011)
2. Barnette, D.W., Grünbaum, B.: On Steinitz's theorem concerning convex 3-polytopes and on some properties of planar graphs. In: Many Facets of Graph Theory, pp. 27–40 (1969)
3. Brandes, U.: Eager *st*-ordering. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 247–256. Springer, Heidelberg (2002)
4. Cheriyan, J., Maheshwari, S.N.: Finding nonseparating induced cycles and independent spanning trees in 3-connected graphs. J. Algorithms 9(4), 507–537 (1988)
5. Curran, S., Lee, O., Yu, X.: Finding four independent trees. SIAM Journal on Computing 35(5), 1023–1058 (2006)

6. de Fraysseix, H., Pach, J., Pollack, R.: Small sets supporting fary embeddings of planar graphs. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC 1988), pp. 426–433 (1988)
7. de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Combinatorica 10(1), 41–51 (1990)
8. Di Battista, G., Tamassia, R., Vismara, L.: Output-sensitive reporting of disjoint paths. Algorithmica 23(4), 302–340 (1999)
9. Hopcroft, J.E., Tarjan, R.E.: Dividing a graph into triconnected components. SIAM J. Comput. 2(3), 135–158 (1973)
10. Huck, A.: Independent trees in planar graphs. Graphs and Combinatorics 15(1), 29–77 (1999)
11. Itai, A., Rodeh, M.: The multi-tree approach to reliability in distributed networks. Information and Computation 79, 43–59 (1988)
12. Kant, G.: Drawing planar graphs using the lmc-ordering. In: Proceedings of the 33th Annual Symposium on Foundations of Computer Science (FOCS 1992), pp. 101–110 (1992)
13. Kant, G.: Drawing planar graphs using the canonical ordering. Algorithmica 16(1), 4–32 (1996)
14. Lovász, L.: Computing ears and branchings in parallel. In: Proceedings of the 26th Annual Symposium on Foundations of Computer Science (FOCS 1985), pp. 464–467 (1985)
15. Mondshein, L.F.: Combinatorial Ordering and the Geometric Embedding of Graphs. PhD thesis, M.I.T. Lincoln Laboratory / Harvard University (1971), Technical Report, available at `http://www.dtic.mil/cgi-bin/GetTRDoc?AD=AD0732882`
16. Mutzel, P.: The SPQR-tree data structure in graph drawing. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 34–46. Springer, Heidelberg (2003)
17. Schmidt, J.M.: Contractions, removals and certifying 3-connectivity in linear time. SIAM Journal on Computing 42(2), 494–535 (2013)
18. Suzuki, H., Takahashi, N., Nishizek, T., Miyano, H., Ueno, S.: An algorithm for tripartitioning 3-connected graphs. Information Processing Society of Japan (IPSJ) 31(5), 584–592 (1990) (in Japanese).
19. Thomassen, C.: Kuratowski's theorem. J. Graph Theory 5(3), 225–241 (1981)
20. Tutte, W.T.: How to draw a graph. Proc. Lond. Math. Soc. 13, 743–767 (1963)
21. Tutte, W.T.: Connectivity in graphs. In: Mathematical Expositions, vo. 15, University of Toronto Press (1966)
22. Wada, K., Kawaguchi, K.: Efficient algorithms for tripartitioning triconnected graphs and 3-edge-connected graphs. In: van Leeuwen, J. (ed.) WG 1993. LNCS, vol. 790, pp. 132–143. Springer, Heidelberg (1994)
23. Whitney, H.: Non-separable and planar graphs. Trans. Amer. Math. Soc. 34(1), 339–362 (1932)
24. Zehavi, A., Itai, A.: Three tree-paths. J. Graph Theory 13(2), 175–188 (1989)