Computing Bounded-Degree Phylogenetic Roots of Disconnected Graphs

Zhi-Zhong Chen *
Department of Mathematical Sciences
Tokyo Denki University
Hatoyama, Saitama 350-0394, Japan

Tatsuie Tsukiji
Department of Information Science
Tokyo Denki University
Hatoyama, Saitama 350-0394, Japan

July 14, 2004

Abstract

The Phylogenetic kth Root Problem (PRk) is the problem of finding a (phylogenetic) tree T from a given graph $G = (V, E)$ such that (1) T has no degree-2 internal nodes, (2) the external nodes (*i.e.* leaves) of T are exactly the elements of V, and (3) $(u, v) \in E$ if and only if the distance between u and v in tree T is at most k, where k is some fixed threshold k. Such a tree T, if exists, is called a *phylogenetic* kth *root* of graph G. The computational complexity of PRk is open, except for $k \leq 4$. Recently, Chen *et al.* investigated PRk under a natural restriction that the maximum degree of the phylogenetic root is bounded from above by a constant. They presented a linear-time algorithm that determines if a given *connected* G has such a phylogenetic kth root, and if so, demonstrates one. In this paper, we supplement their work by presenting a linear-time algorithm for *disconnected* graphs.

Keywords. Phylogeny, phylogenetic root, computational biology, graph power, tree power, graph algorithm.

1 Introduction

The reconstruction of evolutionary history for a set of species from quantitative biological data has long been a popular problem in computational biology. This evolutionary history is typically modeled by an evolutionary tree or *phylogeny*. A phylogeny is a tree where the leaves are labeled by species and each internal node represents a speciation event whereby a hypothetical ancestral species gives rise to two or more child species. Proximity within a phylogeny in general corresponds to similarity in evolutionary characteristics. Both rooted and unrooted trees have been used to describe phylogenies in the literature, although they are practically equivalent. In this paper, we will consider only unrooted phylogenies for the convenience of presentation. Note that each internal node in a phylogeny has at least 3 neighbors.

*Supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan, under Grant No. 14580390.

Many approaches to phylogenetic reconstruction have been proposed in the literature [8]. In particular, Lin *et al.* [4] recently suggested a graph-theoretic approach for reconstructing phylogenies from similarity data. Specifically, interspecies similarity is represented by a graph G where the vertices are the species and the adjacency relation represents evidence of evolutionary similarity. A phylogeny is then reconstructed from G such that the leaves of the phylogeny are labeled by vertices of G (*i.e.* species) and for any two vertices of G, they are adjacent in G if and only if their corresponding leaves in the phylogeny are at most distance k apart, where k is a predetermined proximity threshold. This approach gives rise to the following algorithmic problem [4]:

> PHYLOGENETIC kTH ROOT PROBLEM (PRk):
> Given a graph $G = (V, E)$, find a phylogeny T with leaves labeled by the elements of V such that for each pair of vertices $u, v \in V$, $(u, v) \in E$ if and only if $d_T(u, v) \leq k$, where $d_T(u, v)$ is the number of edges on the path between u and v in T.

Such a phylogeny T (if exists) is called a *phylogenetic* kth *root*, or a kth *root phylogeny*, of graph G. Graph G is called the kth *phylogenetic power* of T. For convenience, we denote the kth phylogenetic power of any phylogeny T as T^k. That is, $T^k = \{(u, v) \mid u \text{ and } v \text{ are leaves of } T \text{ and } d_T(u, v) \leq k\}$. Thus, PR$k$ asks for a phylogeny T such that $G = T^k$.

1.1 Previous Results on PRk

PRk was first studied in [4] where linear-time algorithms for PRk with $k \leq 4$ were proposed. At present, the complexity of PRk with $k \geq 5$ is still unknown.

The hardness of PRk for large k seems to come from the unbounded degree of an internal node in the output phylogeny. On the other hand, in the practice of phylogeny reconstruction, most phylogenies considered are trees of degree 3 [8] because speciation events are usually bifurcating events in the evolutionary process. These motivated Chen *et al.* [2] to consider a restricted version of PRk where the output phylogeny is assumed to have degree at most Δ, for some fixed constant $\Delta \geq 3$. We call this restricted version the DEGREE-Δ PRk and denote it for short as ΔPRk.

Chen *et al.* [2] presented a linear-time algorithm that determines, for any input *connected* graph G and constant $\Delta \geq 3$, if G has a kth root phylogeny with degree at most Δ, and if so, demonstrates one such phylogeny. Unfortunately, their algorithm fails when the input graph G is disconnected. One of their open questions asks for a polynomial-time algorithm for disconnected graphs, because the disconnected case is real in biology.

1.2 Other Problems Related to PRk

A graph G is the kth *power* of a graph H (or equivalently, H is a kth *root* of G), if vertices u and v are adjacent in G if and only if they are at most distance k apart in H. An important special case of graph power/root problems is the TREE kTH ROOT PROBLEM (TRk): Given a graph $G = (V, E)$, we wish to find a tree $T = (V, E_T)$ such that $(u, v) \in E$ if and only if $d_T(u, v) \leq k$. If T exists, then it is called a *tree* kth *root*, or a kth *root tree*, of graph G. There is rich literature on graph roots and powers (see [1, Section 10.6] for an overview), but few results on phylogenetic/tree roots/powers. It is NP-complete to recognize a graph power [6]; nonetheless, we can determine if a graph has a kth root tree, for any fixed k, in cubic time [3]. In particular, determining if a graph has a tree square

root can be done in linear time [5]. Moreover, Nishimura *et al.* [7] presented a cubic time algorithm for a variant of PRk with $k \leq 4$, where internal nodes of the output phylogeny are allowed to have degree 2.

1.3 Our Contribution

Our result is a linear-time algorithm that determines, for any input *disconnected* graph G and constant $\Delta \geq 3$, if G has a kth root phylogeny with degree at most Δ, and if so, demonstrates one such phylogeny. This answers an open question in [2]. Combining this algorithm with the algorithm in [2] for connected graphs, we obtain the first linear-time algorithm for ΔPRk for any constants $\Delta \geq 3$ and $k \geq 2$. Our algorithm is complicated and it is based on hidden structures of phylogenetic kth roots of disconnected graphs. Moreover, the algorithm needs a linear-time subroutine for solving a certain optimization problem on each connected component of the input disconnected graph. The subroutine is obtained by nontrivially refining the algorithm in [2].

1.4 Organization of the Paper

We introduce some notations and definitions in the next section. Our linear-time algorithm for ΔPRk is presented in Sections 3 and 4.

2 Preliminaries

We employ standard terminologies in graph theory. In particular, the subgraph of a graph G induced by a vertex set U of G is denoted by $G[U]$, the degree of a vertex v in G is denoted by $deg_G(v)$, and the distance between two vertices u and v in G is denoted by $d_G(u, v)$. Moreover, for a set W of vertices in a graph $G = (V, E)$, we write $G - W$ for $G[V - W]$. Furthermore, in a rooted tree, each vertex is both an ancestor and a descendant of itself.

For clarity, if $G = (V, E)$ is a graph and $T = (V_T, E_T)$ is a kth root phylogeny of G for some k, then we call the elements of V *vertices* and call those of V_T *nodes*.

In the remainder of this section, fix a graph $G = (V, E)$ and two integers $k \geq 4$ and $\Delta \geq 3$. A *degree-Δ kth root phylogeny* ((Δ, k)-phylogeny for short) of G is a kth root phylogeny T of G such that the maximum degree of a node in T is at most Δ.

A *degree-Δ kth root quasi-phylogeny* ((Δ, k)-QP for short) of G is a tree Q satisfying the following conditions:

- Each vertex of G is a leaf of Q and appears in Q exactly once. For convenience, we call the leaves of Q that are also vertices of G *true leaves* of Q, and call the other leaves of Q *false leaves* of Q.
- The degree of each node in Q is at most Δ.
- For every two vertices u and v in G, u and v are adjacent in G if and only if $d_Q(u, v) \leq k$.
- For each node x of Q that is a degree-2 node or a false leaf in Q, it holds that $\min_{v \in V} d_Q(x, v) \geq \lfloor \frac{k}{2} \rfloor$.
- If Q has no false leaf, then it has at least one node x such that $2 \leq deg_Q(x) \leq \Delta - 1$ and $\min_{v \in V} d_Q(x, v) \geq \lfloor \frac{k}{2} \rfloor$.

The *cost* of Q is $\max\{1, a+2b\}$, where a is the number of degree-2 nodes in Q and b is the number of false leaves in Q. Q is an *optimal* (Δ, k)-QP of G if its cost is minimized over all (Δ, k)-QPs of G.

Lemma 2.1 *Suppose that $G = (V, E)$ is a connected graph. Let Q be an optimal (Δ, k)-QP of G. Then, the following hold:*

1. *Q has no node x with $\min_{v \in V} d_Q(x, v) > \lfloor \frac{k}{2} \rfloor$.*
2. *For each node x with $deg_Q(x) = 2$ or $deg_Q(x) > 3$, each connected component of $Q - \{x\}$ contains at least one true leaf of Q.*

Proof. We prove the two statements separately as follows.

Statement 1. If x were a false leaf of Q with $\min_{v \in V} d_Q(x, v) > \lfloor \frac{k}{2} \rfloor$, then the removal of x from Q would result in a new (Δ, k)-QP of G whose cost is smaller than that of Q, a contradiction. So, for every false leaf x of Q, $\min_{v \in V} d_Q(x, v) \leq \lfloor \frac{k}{2} \rfloor$. In turn, for every internal node x of Q such that one connected component of $Q - \{x\}$ contains all true leaves of Q, it holds that $\min_{v \in V} d_Q(x, v) \leq \lfloor \frac{k}{2} \rfloor$.

Now, it remains to consider those internal nodes x of Q such that no connected component of $Q - \{x\}$ contains all true leaves of Q. If among these nodes, there were one x with $\min_{v \in V} d_Q(x, v) > \lfloor \frac{k}{2} \rfloor$, then G would have no edge (u, v) such that u and v belong to different connected components of $Q - \{x\}$, contradicting the connectivity of G.

Statement 2. Let x be a node of Q with $deg_Q(x) = 2$ or $deg_Q(x) > 3$. For a contradiction, assume that some connected component C of $Q - \{x\}$ contains no true leaf of Q. If $deg_Q(x) > 3$, then the removal of C from Q results in a new (Δ, k)-QP of G whose cost is smaller than that of Q, a contradiction. If $deg_Q(x) = 2$, then by Statement 1, $\min_{v \in V} d_Q(x, v) < \lfloor \frac{k}{2} \rfloor$, a contradiction. □

We classify (Δ, k)-QPs Q into four types as follows.

- Q is *helpful* if it has at most one degree-2 node and has no false leaf.
- Q is *moderate* if it has no degree-2 node but has exactly one false leaf.
- Q is *troublesome* if it has at least two degree-2 nodes but has no false leaf.
- Q is *dangerous* if it has at least one false leaf and the total number of false leaves and degree-2 nodes in Q is at least 2.

A (Δ, k)-QP Q is *unhelpful* if it is not helpful.

For a (Δ, k)-QP Q, we define its *port nodes* as follows. If Q is not helpful, then its port nodes are its false leaves and degree-2 nodes. If Q is helpful and has no degree-2 node, then its port nodes are those nodes x with $\min_{v \in V} d_Q(x, v) \geq \lfloor \frac{k}{2} \rfloor$. If Q is helpful and has a degree-2 node, then it has only one port node, namely, its unique degree-2 node.

A *nonport node* of a (Δ, k)-QP Q is a node of Q that is not a port node of Q.

3 Algorithm for Bounded-Degree PRk

Throughout this section, fix two integers $k \geq 4$ and $\Delta \geq 3$. This section presents a linear-time algorithm for solving ΔPRk.

Let $G = (V, E)$ be the input graph. We assume that G is disconnected; otherwise, the linear-time algorithm in [2] solves the problem. Let $G_1, \ldots, G_\ell$ be the connected components of G. For each integer with $1 \leq i \leq \ell$, let V_i be the vertex set of G_i.

In the next section, we will show the following lemma:

Lemma 3.1 *For every $i \in \{1, \ldots, \ell\}$, we can decide whether G_i has a (Δ, k)-QP, in $O(|V_i|)$ time. Moreover, if G_i has a (Δ, k)-QP, then we can compute an optimal (Δ, k)-QP of G_i in $O(|V_i|)$ time.*

Lemma 3.2 *If for some $i \in \{1, \ldots, \ell\}$, G_i has no (Δ, k)-QP, then G has no (Δ, k)-phylogeny.*

Proof. Suppose that G has a (Δ, k)-phylogeny T. Fix an $i \in \{1, \ldots, \ell\}$. Let Y_i be the set of all internal nodes y of T such that there is a vertex $u \in V_i$ with $d_T(u, y) \leq \lfloor \frac{k}{2} \rfloor$. Obviously, $T[V_i \cup Y_i]$ is a (Δ, k)-QP of G_i. □

By Lemmas 3.1 and 3.2, we may assume that for each $i \in \{1, \ldots, \ell\}$, G_i has a (Δ, k)-QP. For each $i \in \{1, \ldots, \ell\}$, let Q_i be the optimal (Δ, k)-QP of G_i computed in Lemma 3.1.

Lemma 3.3 *Suppose that G has a (Δ, k)-phylogeny. Then, G has a (Δ, k)-phylogeny T such that $Q_1, \ldots, Q_\ell$ all are subtrees of T.*

Proof. Let T be a (Δ, k)-phylogeny of G. For each $i \in \{1, \ldots, \ell\}$, let Y_i be as in the proof of Lemma 3.2. Recall that $T[V_i \cup Y_i]$ is a (Δ, k)-QP of G_i. Moreover, for every pair (i, j) with $1 \leq i \neq j \leq \ell$, $Y_i \cap Y_j = \emptyset$.

Consider the integer $i \in \{1, \ldots, \ell\}$ such that $Q_1, \ldots, Q_{i-1}$ are subtrees of T but Q_i is not. If no such i exists, then T is as required. So, assume that i exists. Let $F_1, \ldots, F_h$ be the connected components of $T - (V_i \cup Y_i)$. For each $j \in \{1, \ldots, h\}$, F_j has exactly one node z_j with $deg_{F_j}(z_j) < deg_T(z_j)$, and each leaf of F_j is a vertex in $V - V_i$. Moreover, the minimum distance from z_j to a leaf in F_j is at least $\lceil \frac{k}{2} \rceil$, because $\min_{v \in V_i} d_T(z_j, v) = \lfloor \frac{k}{2} \rfloor + 1$. Let $Z = \{z_1, \ldots, z_h\}$. Define a function $f : Z \to Y_i$ as follows. For each $j \in \{1, \ldots, h\}$, let $f(z_j)$ be the neighbor of z_j in T that is not in F_j. Let $X = \{f(z_j) \mid 1 \leq j \leq h\}$.

We claim that the cost of the (Δ, k)-QP $T[V_i \cup Y_i]$ of G_i is at most h. To see this claim, first note that X contains all false leaves and all degree-2 nodes of $T[V_i \cup Y_i]$. Moreover, for each degree-2 node x of $T[V_i \cup Y_i]$, there is at least one $z_j \in Z$ with $f(z_j) = x$. Furthermore, for each false leaf x of $T[V_i \cup Y_i]$, there are at least two $z_j \in Z$ with $f(z_j) = x$. Hence, the claim holds. By the claim, the cost of the optimal (Δ, k)-QP Q_i of G_i is at most h.

Now, we use Q_i and $F_1, \ldots, F_h$ to obtain a new (Δ, k)-phylogeny T_i of G, by performing the following steps:

1. Let $x_1, \ldots, x_a$ be the degree-2 nodes in Q_i, and let $x_{a+1}, \ldots, x_{a+b}$ be the false nodes in Q_i.
2. If $a > 0$ or $b > 0$, then set $c = a + 2b$; otherwise, set $c = 1$ and let x_1 be an (arbitrarily chosen) port node of Q_i. (Comment: c is the cost of Q_i and $h \geq c$.)
3. For all j with $1 \leq j \leq c - 2b$, add edge (x_j, z_j).

4. For all j with $1 \leq j \leq b$, add edges $(x_{c-2b+j}, z_{c-2b+2j-1})$ and $(x_{c-2b+j}, z_{c-2b+2j})$.

5. If $h > c$, then perform the following steps:

   (a) Delete the edge between x_{c-b} and z_c.

   (b) Introduce $h-c$ new nodes $y_1, \ldots, y_{h-c}$, and connect them into a path from y_1 to y_{h-c}.

   (c) For all i with $1 \leq i \leq h-c$, add edge (y_i, z_{c+i-1}).

   (d) Add edges (y_1, x_{c-b}) and (y_{h-c}, z_h).

Obviously, T_i is a (Δ, k)-phylogeny of G and Q_i is a subtree of T_i. Moreover, for every j with $1 \leq j \leq i-1$, Q_j remains to be a subtree of T_i, because $Y_i \cap Y_j = \emptyset$ and Q_j is a subtree of $T[V_j \cup Y_j]$ by Statement 1 in Lemma 2.1. Thus, T_i is a (Δ, k)-phylogeny of G such that for all $j \in \{1, \ldots, i\}$, Q_j is a subtree of T_i. Therefore, by our choice of i, we can repeat the above argument to finally obtain a (Δ, k)-phylogeny T_ℓ of G such that $Q_1, \ldots, Q_\ell$ are subtrees of T_ℓ. □

In the remainder of this section, a (Δ, k)-phylogeny of G always means one in which $Q_1, \ldots, Q_\ell$ are subtrees. By Lemma 3.3, we lose no generality. For convenience, we call $Q_1, \ldots, Q_\ell$ the *unitary* (Δ, k)-QPs.

Let T be a (Δ, k)-phylogeny T of G. A *junction node* of T is a node x of T such that no unitary (Δ, k)-QP contains x. A node x of T is *over-connected*, if it satisfies one of the following conditions:

(1) $deg_T(x) > 3$ and x is a junction node of T.

(2) $deg_T(x) > 3$ and x is a port node of some unhelpful Q_i $(1 \leq i \leq \ell)$.

(3) x is a nonport node of some unhelpful Q_i $(1 \leq i \leq \ell)$ and $deg_T(x) > deg_{Q_i}(x)$.

A helpful Q_i $(1 \leq i \leq \ell)$ is *mis-connected* in T, if (i) at least one nonport node of Q_i is adjacent to a node outside Q_i in T, or (ii) there are two or more nodes x outside Q_i such that x is adjacent to a node of Q_i in T.

A (Δ, k)-phylogeny T of G is *canonical*, if it has no over-connected node and no helpful Q_i $(1 \leq i \leq \ell)$ is mis-connected in T.

Lemma 3.4 *If G has a (Δ, k)-phylogeny, then it has a canonical one.*

PROOF. Let T be a (Δ, k)-phylogeny of G. Suppose that T has an over-connected node x. We distinguish three cases as follows.

Case 1: x satisfies Condition (1) above. Let $z_1, \ldots, z_h$ be the neighbors of x in T. Let $b = h-2$. We modify T by performing the following four steps:

1. Delete the edges $(x, z_1), \ldots, (x, z_b)$.

2. Introduce $b-1$ new junction nodes $w_1, \ldots, w_{b-1}$, and connect them into a path from w_1 to w_{b-1}.

3. For each $j \in \{1, \ldots, b-1\}$, add edge (w_j, z_j).

4. Add edges (w_1, x) and (w_{b-1}, z_b).

The modification does not decrease the distance between each pair of leaves in T. So, after the modification, T remains to be a (Δ, k)-phylogeny of G.

Case 2: x satisfies Condition (2) above. Let Q_i $(1 \leq i \leq \ell)$ be the unitary (Δ, k)-QP containing x. Let $z_1, \ldots, z_h$ be the neighbors of x outside R_i in T. By Statement 1 in Lemma 2.1, $z_1, \ldots, z_h$ are junction nodes of T. If x is a degree-2 node of Q_i, then let $b = h$; otherwise (x is a false leaf of Q_i), let $b = h - 1$. We modify T by performing the four steps in Case 1. After the modification, T remains to be a (Δ, k)-phylogeny of G, as in Case 1.

Case 3: x satisfies Condition (3) above. Let Q_i $(1 \leq i \leq \ell)$ be the unitary (Δ, k)-QP containing x. Let $z_1, \ldots, z_h$ be the neighbors of x outside R_i in T. Let y be an (arbitrarily chosen) port node of R_i. Let z_{h+1} be an (arbitrarily chosen) neighbor of y outside R_i in T. By Statement 1 in Lemma 2.1, $z_1, \ldots, z_{h+1}$ are junction nodes of T. We modify T as follows:

1. Delete the edges $(x, z_1), \ldots, (x, z_h), (y, z_{h+1})$. (Comment: After this step, $\min_{v \in V} d_T(z_j, v) \geq \lceil \frac{k}{2} \rceil$ for all $j \in \{1, \ldots, h+1\}$. This follows from Statement 1 in Lemma 2.1.)

2. Introduce h new junction nodes $w_1, \ldots, w_h$, and connect them into a path from w_1 to w_h.

3. For each $j \in \{1, \ldots, h\}$, add edge (w_j, z_j).

4. Add edges (w_1, y) and (w_h, z_{h+1}). (Comment: After this step, T becomes a (Δ, k)-phylogeny of G. This follows from the comment on Step 1.)

Obviously, after the modification in each case above, $Q_1, \ldots, Q_\ell$ remain to be subtrees of T (because each edge deleted is adjacent to a junction node of T before the modification), and x becomes not over-connected in T while no node newly becomes over-connected in T. So, we can repeat modifying T until it has no over-connected node.

Next, suppose that some helpful Q_i $(1 \leq i \leq \ell)$ is mis-connected in T. Let $z_1, \ldots, z_h$ be the nodes outside Q_i that are adjacent to nodes of Q_i in T. By Statement 1 in Lemma 2.1, $z_1, \ldots, z_h$ are junction nodes of T. Let x be an (arbitrarily chosen) port node of Q_i. We modify T as follows.

1. For each $j \in \{1, \ldots, h\}$, delete edge (z_j, y_j) where y_j is the node of Q_i adjacent to z_j in T. (Comment: After this step, $\min_{v \in V} d_T(z_j, v) \geq \lceil \frac{k}{2} \rceil$ for all $j \in \{1, \ldots, h+1\}$. This follows from Statement 1 in Lemma 2.1.)

2. If $h = 1$, then add edge (z_1, x).

3. If $h > 1$, then set $h = b$ and perform the last three steps in Case 1.

Obviously, after the modification, $Q_1, \ldots, Q_\ell$ remain to be subtrees of T (because $z_1, \ldots, z_h$ are junction nodes), T remains to be a (Δ, k)-phylogeny of G (by the comment on Step 1), T remains to have no over-connected node, and Q_i becomes not mis-connected in T while no helpful $Q_j \neq Q_i$ newly becomes mis-connected in T. So, we can repeat modifying T until T becomes canonical. □

In the remainder of this section, a (Δ, k)-phylogeny of G always means a canonical one. By Lemma 3.4, we lose no generality.

3.1 The Case where k is Odd

Throughout this subsection, we assume that k is odd. A *double* (Δ, k)-QP is a tree $T_{i,j}$ obtained by combining two helpful unitary (Δ, k)-QPs Q_i and Q_j as follows:

1. Select a port node x_i of Q_i, and select a port node x_j of Q_j.
2. Introduce a junction node y, and connect it to both x_i and x_j.

Note that $T_{i,j}$ has exactly one degree-2 node (namely, the junction node y) but has no false leaf. So, $T_{i,j}$ is a helpful (Δ, k)-QP of $G[V_i \cup V_j]$. Moreover, the minimum distance from y to a true leaf in $T_{i,j}$ is exactly $\lfloor \frac{k}{2} \rfloor + 1$ (cf. Statement 1 in Lemma 2.1).

Lemma 3.5 *Suppose that each Q_i ($1 \leq i \leq \ell$) is helpful or moderate. Then, G has a (Δ, k)-phylogeny if and only if $\ell \geq 2b+3$, where b is the number of moderate (Δ, k)-QPs among $Q_1, \ldots, Q_\ell$.*

Proof. We prove the two directions separately as follows.

($\Longrightarrow$) Suppose that G has a (Δ, k)-phylogeny T. For each moderate Q_i, let Q'_i be the tree obtained from Q_i by deleting its unique false leaf. Let $\mathcal{T}$ be the tree obtained by modifying T as follows:

1. For each helpful Q_i ($1 \leq i \leq \ell$), merge Q_i into a super-node s_i.
2. For each moderate Q_i ($1 \leq i \leq \ell$), merge Q'_i into a super-node s'_i.

Obviously, the leaves of $\mathcal{T}$ are exactly the super-nodes. So, $\mathcal{T}$ has exactly ℓ leaves. Let c be the number of internal nodes of $\mathcal{T}$. Let $m_{\mathcal{T}}$ be the number of edges in $\mathcal{T}$. Note that the false leaf x_i of each moderate Q_i remains to be an internal node in $\mathcal{T}$, no neighbor of x_i in $\mathcal{T}$ is the false leaf x_j of another moderate Q_j in $\mathcal{T}$, and no neighbor of x_i in $\mathcal{T}$ is a super-node s_j corresponding to a helpful Q_j (cf. Statement 1 in Lemma 2.1). This implies that $m_{\mathcal{T}} \geq 3b + (\ell - b)$. Trivially, $m_{\mathcal{T}} = \ell + c - 1$, and $m_{\mathcal{T}} = \frac{3c+\ell}{2}$ because the degree of each internal node in $\mathcal{T}$ is exactly 3 (by the canonicity of T). Therefore, $\ell \geq 2b + 3$.

($\Longleftarrow$) Suppose that $\ell \geq 2b + 3$. By renumbering if necessary, we may assume that $Q_1, \ldots, Q_b$ are moderate. For each $i \in \{1, \ldots, b\}$, let y_i be the false leaf of Q_i. For each $i \in \{b+1, \ldots, \ell\}$, let z_i be an (arbitrarily chosen) port node of Q_i. We can connect $Q_1, \ldots, Q_\ell$ into a (Δ, k)-phylogeny of G as follows.

1. Introduce $\ell - b - 2$ junction nodes $x_1, \ldots, x_{\ell-b-2}$.
2. For each i with $1 \leq i \leq b$, add edges (y_i, x_i) and (y_i, x_{i+1}).
3. Add edges (x_1, z_{b+1}), (x_1, z_{b+2}), $(x_{\ell-b-2}, z_{\ell-1})$, and $(x_{\ell-b-2}, z_\ell)$. (Comment: If $\ell - b = 3$, then only three edges are added here.)
4. For each i with $b + 1 \leq i \leq \ell - b - 3$, add edge (x_i, x_{i+1}).
5. For each i with $2 \leq i \leq \ell - b - 3$, add edge (x_i, z_{b+i+1}).

□

In the sequel, we assume that at least one Q_i $(1 \leq i \leq \ell)$ is troublesome or dangerous (since otherwise Lemma 3.5 solves the problem).

Let T be a (Δ, k)-phylogeny of G. For each dangerous Q_i $(1 \leq i \leq \ell)$, we say that a false leaf x of Q_i is *active* in T, if no connected component of $T - \{x\}$ is a double (Δ, k)-QP. A dangerous Q_i $(1 \leq i \leq \ell)$ is *active* in T if at least one false leaf of Q_i is active in T.

Lemma 3.6 *Suppose that G has a (Δ, k)-phylogeny. Then, G has a (Δ, k)-phylogeny T such that no dangerous Q_i $(1 \leq i \leq \ell)$ is active in T.*

Proof. Let T be a (Δ, k)-phylogeny of G. Consider the integer $i \in \{1, \ldots, \ell\}$ such that no dangerous Q_j with $1 \leq j \leq i-1$ is active in T but Q_i is both dangerous and active in T. If no such i exists, then T is as required. So, assume that i exists. Let $x_1, \ldots, x_h$ be the false leaves of Q_i that are active in T.

Root T at a true leaf of Q_i. A *junction descendant* of a node x in T is a descendant of x in T that is a junction node. Let y_1 and y_2 be the children of x_1 in T. Since the minimum distance from x_1 to a true leaf of Q_i in T is exactly $\lfloor \frac{k}{2} \rfloor$ (cf. Lemma 2.1), the minimum distance from each y_j $(1 \leq j \leq 2)$ to a leaf descendant of y_j in T is at least $\lceil \frac{k}{2} \rceil$. Thus, by Statement 1 in Lemma 2.1, both y_1 and y_2 are junction nodes in T.

Let y_3 be a junction descendant of y_1 in T such that $d_T(y_1, y_3) \geq d_T(y_1, y_3')$ for all junction descendants y_3' of y_1 in T. Then, the subtree of T rooted at y_3 must be a double (Δ, k)-QP. So, $y_3 \neq y_1$, since x_1 is active in T. Let z be the parent of y_3 in T. It is possible that $z = y_1$. Let y_4 be the other child of z than y_3. We distinguish two cases as follows.

Case 1: y_4 is a junction node in T. Then, by the choice of y_3, the subtree of T rooted at y_4 is also a double (Δ, k)-QP. So, we can modify T to obtain a new (Δ, k)-phylogeny P_1 of G by deleting edges (z, y_3) and (x_1, y_2) and adding edges (x_1, y_3) and (z, y_2). Obviously, x_1 becomes inactive in P_1, and each dangerous Q_j with $1 \leq j \leq i-1$ remains inactive in P_1 (even if z is a false leaf of Q_j).

Case 2: y_4 is not a junction node in T. Let $Q_{i'}$ be the unitary (Δ, k)-QP containing y_4. Obviously, $i' \neq i$. By the choice of y_3, the subtree of T rooted at y_4 is a subtree of $Q_{i'}$. Moreover, if $Q_{i'}$ contains z, then either z is not a false leaf of $Q_{i'}$ or $Q_{i'}$ is moderate. Furthermore, if $Q_{i'}$ does not contain z, then by Statement 1 in Lemma 2.1, z cannot be a false leaf of a dangerous unitary (Δ, k)-QP. So, no matter whether $Q_{i'}$ contains z or not, z cannot be a false leaf of a dangerous unitary (Δ, k)-QP. Thus, we can modify T to obtain a new (Δ, k)-phylogeny P_1 of G as in Case 1, and P_1 has the same properties as stated in Case 1.

By Cases 1 and 2, we can always modify T to obtain a new (Δ, k)-phylogeny P_1 of G such that x_1 becomes inactive in P_1 and each dangerous Q_j with $1 \leq j \leq i-1$ remains inactive in P_1. Since the modification only changes the subtree of T rooted at x_1 and the subtrees of T rooted at two different false leaves of Q_i are disjoint, we can then similarly modify P_1 to obtain P_2, further modify P_2 to obtain P_3, and so on, in such a way that for all $i' \in \{2, 3, \ldots, h\}$, $x_1, \ldots, x_{i'}$ are inactive in $P_{i'}$ and each dangerous Q_j with $1 \leq j \leq i-1$ remains inactive in $P_{i'}$.

Let $T_i = P_h$. Then, no dangerous Q_j with $1 \leq j \leq i$ is active in T_i. Therefore, by our choice of i, we can repeat the above argument to finally obtain a (Δ, k)-phylogeny T' of G such that no dangerous unitary (Δ, k)-QP is active in T'. □

Let I be the set of all $i \in \{1, \ldots, \ell\}$ such that Q_i is dangerous. For each $i \in I$, let t_i be the number of false leaves in Q_i. Let $t = \sum_{i \in I} t_i$. By Lemma 3.6, if G has a (Δ, k)-phylogeny, then there

are at least $2t$ helpful unitary (Δ, k)-QPs. So, if there are less than $2t$ helpful unitary (Δ, k)-QPs, then G has no (Δ, k)-phylogeny. In the sequel, we assume that there are at least $2t$ helpful unitary (Δ, k)-QPs. Without loss of generality, we may assume that $Q_1, \ldots, Q_{2t}$ are helpful.

We connect $Q_1, \ldots, Q_{2t}$ to the dangerous unitary (Δ, k)-QPs as follows.

1. Introduce t junction nodes $x_1, \ldots, x_t$, and construct a one-to-one correspondence between them and the t false leaves of the dangerous unitary (Δ, k)-QPs.

2. For each $i \in \{1, \ldots, t\}$, add an edge from x_i to its corresponding false leaf, add an edge from x_i to an (arbitrarily chosen) port node of Q_{2i-1}, and add an edge from x_i to an (arbitrarily chosen) port node of Q_{2i}.

The above modification extends each dangerous unitary (Δ, k)-QP Q_i to a troublesome (Δ, k)-QP R_i. For convenience, let $R_i = Q_i$ for each $i \in \{2t+1, \ldots, \ell\}$ such that Q_i is not dangerous.

Now, we are left with $R_{2t+1}, \ldots, R_\ell$; none of them is dangerous. Let τ be the number of troublesome (Δ, k)-QPs among $R_{2t+1}, \ldots, R_\ell$. Note that $\tau = |i \in \{1, \ldots, \ell\} \mid Q_i$ is troublesome or dangerous$\}$. So, $\tau \geq 1$. Without loss of generality, we may assume that $R_{2t+1}, \ldots, R_{2t+\tau}$ are troublesome.

By Lemma 3.6, if G has a (Δ, k)-phylogeny, then it has one in which $R_{2t+1}, \ldots, R_\ell$ are subtrees. So, in the remainder of this section, a (Δ, k)-phylogeny of G always means one in which $R_{2t+1}, \ldots, R_\ell$ are subtrees.

A *bridging node* in a (Δ, k)-phylogeny T of G is a node x of T such that no R_i with $2t+1 \leq i \leq \ell$ contains x. For each (Δ, k)-phylogeny T of G and for each R_i with $2t+1 \leq i \leq \ell$, each degree-2 node x of R_i is adjacent to exactly one bridging node y in T (by the canonicity of T); we call y the *bridging neighbor* of x in T.

For each (Δ, k)-phylogeny T of G, let $\mathcal{M}(T)$ denote the tree obtained by modifying T by merging each R_i with $2t+1 \leq i \leq \ell$ into a super-node. For convenience, we abuse the notation to let each R_i also denote the super-node of $\mathcal{M}(T)$ corresponding to R_i. Note that each bridging node of T remains to be an internal node in $\mathcal{M}(T)$ and the leaves of $\mathcal{M}(T)$ one-to-one correspond to the helpful unitary (Δ, k)-QPs among $R_{2t+1}, \ldots, R_\ell$. Moreover, by the canonicity of T and Statement 1 in Lemma 2.1, no two super-nodes can be adjacent in $\mathcal{M}(T)$.

Lemma 3.7 *If G has a (Δ, k)-phylogeny, then it has one T such that there is a path q in $\mathcal{M}(T)$ on which $R_{2t+1}, \ldots, R_{2t+\tau}$ appear.*

Proof. Let T be a (Δ, k)-phylogeny of G. Since the leaves of $\mathcal{M}(T)$ one-to-one correspond to the helpful unitary (Δ, k)-QPs among $R_{2t+1}, \ldots, R_\ell$, we have $2t + \tau < \ell$. Root T at a true leaf r of R_ℓ.

For each troublesome R_i, let x_i be the node of R_i closest to r in T. By the canonicity of T, x_i is a degree-2 node of R_i and the two children of x_i in T are nodes of R_i. Moreover, for each degree-2 node $y \neq x_i$ of R_i, the bridging neighbor of y in T is a child of y in T; for clarity, we call this child the *bridging child* of y in T. We say that a degree-2 node $y \neq x_i$ of R_i is *unsettled* in T, if the subtree of T rooted at the bridging child of y contains at least one of $R_{2t+1}, \ldots, R_{2t+\tau}$ as a subtree. A troublesome R_i is *unsettled* in T if it has two or more unsettled degree-2 nodes.

If no troublesome R_i is unsettled in T, then by the canonicity of T, T is as required. Otherwise, we can keep modifying T as follows, until no troublesome R_i is unsettled in T:

1. Choose an unsettled troublesome R_i such that $d_T(r, x_i) \leq d_T(r, x_j)$ for every unsettled troublesome R_j.
2. Let $y_1, \ldots, y_h$ be the unsettled degree-2 nodes of R_i.
3. For each $j \in \{1, \ldots, h\}$, let y'_j be the bridging child of y_j in T.
4. For each $j = 2, 3, \ldots, h$ (in this order), perform the following steps:
   (a) Find a troublesome $R_{i'}$ such that $x_{i'}$ is a descendant of y'_1 in T and there is no troublesome $R_{i''}$ with $i'' \neq i'$ such that $x_{i''}$ is a descendant of $x_{i'}$ in T.
   (b) Select a degree-2 node z of $R_{i'}$ with $z \neq x_{i'}$, and let z' be the bridging child of z in T.
   (c) Delete edges (z, z') and (y_j, y'_j), and add edges (z, y'_j) and (y_j, z'). (Comment: After this step, each $R_{i''}$ with $2t + 1 \leq i'' \leq \ell$ remains to be a subtree of T. Moreover, y_j becomes not unsettled and $y_1, \ldots, y_{j-1}$ remain not unsettled. Furthermore, if a troublesome $R_{i''}$ with $d_T(r, x_i) \leq d_T(r, x_{i''})$ is not unsettled before this step, then it remains not unsettled after this step.)

□

Lemma 3.8 *If G has a (Δ, k)-phylogeny, then it has one T such that some path q in $\mathcal{M}(T)$ satisfies the following three conditions:*

1. *$R_{2t+1}, \ldots, R_{2t+\tau}$ and exactly $\tau - 1$ bridging nodes appear on q.*
2. *No two bridging nodes on q are adjacent in T.*
3. *For each bridging node x on q, there is a helpful unitary (Δ, k)-QP R_i such that x is adjacent to a port node of R_i in T.*

PROOF. Let T be a (Δ, k)-phylogeny of G. By Lemma 3.7, we may assume that there is a path p in $\mathcal{M}(T)$ on which $R_{2t+1}, \ldots, R_{2t+\tau}$ appear. We may further assume that neither endpoint of p is a bridging node. By renumbering if necessary, we may assume that R_{2t+1} and $R_{2t+\tau}$ are the endpoints of p and $R_{2t+1}, \ldots, R_{2t+\tau}$ appear on p in this order.

Root T at a true leaf r of R_{2t+1}. A *bridging descendant* of a node x in T is a descendant of x in T that is a bridging node. For each troublesome $R_i \neq R_{2t+1}$, let x_i be the node of R_i closest to r in T. By the canonicity of T, x_i is a degree-2 node of R_i. Moreover, for each degree-2 node $y \neq x_i$ of R_i, the bridging neighbor of y in T is a child of y in T; for clarity, we call this child the *bridging child* of y in T.

For each $i \in \{2t+1, \ldots, 2t+\tau-1\}$, let y_i be the degree-2 node of R_i that is an ancestor of x_{i+1} in T. By the canonicity of T, y_i is a degree-2 node of R_i. In turn, by Statement 1 in Lemma 2.1, y_i cannot be the parent of x_{i+1} in T. For each $i \in \{2t+1, \ldots, 2t+\tau-1\}$, let x'_{i+1} be the parent of x_{i+1} in T, and let y'_i be the child of y_i in T that is an ancestor of x_{i+1} in T. Possibly, $y'_i = x'_{i+1}$.

Fix a degree-2 node $z \neq x_{2t+\tau}$ of $R_{2t+\tau}$. Let z_1 be the bridging child of z in T. We can modify T by performing the following steps for $i = 2t+2, \ldots, 2t+\tau$ (in this order):

1. Find a bridging descendant z_2 of z_1 in T such that $d_T(z_1, z_2) \geq d_T(z_1, z_3)$ for every bridging descendant z_3 of z_1 in T. (Comment: Possibly, $z_1 = z_2$. Moreover, the subtree of T rooted at z_2 must be a double (Δ, k)-QP.)

2. Let z_3 be the parent of z_2 in T, and let z_4 and z_5 be the children of z_2 in T. (Comment: Possibly, $z_3 = z_1$ or $z_3 = z$.)

3. Delete edges (y_{i-1}, y'_{i-1}), (x'_i, x_i), (z_3, z_2) and (z_2, z_5), and further add edges (y_{i-1}, z_2), (z_2, x_i), (z_3, y'_{i-1}) and (x'_i, z_5). (Comment: Before and after this step, y'_{i-1}, x'_i, and z_2 are bridging nodes, and each of x_i, y_{i-1}, and z_5 is a degree-2 node of an R_j with $2t+1 \leq j \leq \ell$. So, after this step, $R_{2t+1}, \ldots, R_\ell$ remain to be subtrees of T and T remains to be a (Δ, k)-phylogeny of G.)

Obviously, after the above modification, the path q from the super-node corresponding to R_{2t+1} to the super-node corresponding to $R_{2t+\tau}$ in $\mathcal{M}(T)$ satisfies the conditions in the lemma. □

In the remainder of this section, a (Δ, k)-phylogeny of G always means one T such that some path q in $\mathcal{M}(T)$ satisfies the three conditions in Lemma 3.8. We call q the *spine* of $\mathcal{M}(T)$. The following corollary shows that it does not matter in which order $R_{2t+1}, \ldots, R_{2t+\tau}$ appear on the spine.

Corollary 3.9 *Let T be a (Δ, k)-phylogeny of G. Then, for every pair (R_i, R_j) of troublesome (Δ, k)-QPs, there is another (Δ, k)-phylogeny T' of G such that the spine of $\mathcal{M}(T')$ can be obtained from that of $\mathcal{M}(T)$ by exchanging the positions of R_i and R_j.*

Proof. The corollary is obvious, if $\tau \leq 2$. So, assume that $\tau \geq 3$.

Let q be the spine of $\mathcal{M}(T)$. Consider a pair (R_i, R_j) of troublesome (Δ, k)-QPs. If R_i is not an endpoint of q, then let x_i and y_i be the two degree-2 nodes of R_i whose bridging neighbors are on q; otherwise, let x_i be the degree-2 node of R_i whose bridging neighbor is on q, and let y_i be a degree-2 node of R_i with $y_i \neq x_i$. Let x'_i (respectively, y'_i) be the bridging neighbor of x_i (respectively, y_i) in T. Define x_j, y_j, x'_j, and y'_j similarly. Possibly, $\{x'_i, y'_i\} \cap \{x'_j, y'_j\} \neq \emptyset$.

We obtain T' by deleting edges (x_i, x'_i), (y_i, y'_i), (x_j, x'_j), and (y_j, y'_j), and adding edges (x_i, x'_j), (y_i, y'_j), (x_j, x'_i), and (y_j, y'_i). Obviously, T' is as required. □

The following corollary is obvious and shows that it does not matter via which degree-2 nodes each troublesome R_i is connected to the spine.

Corollary 3.10 *Let T be a (Δ, k)-phylogeny of G. Then, for every troublesome R_i and for every pair (x_1, x_2) of degree-2 nodes of R_i, we can obtain another (Δ, k)-phylogeny T' of G by deleting edges (x_1, y_1) and (x_2, y_2) and adding edges (x_1, y_2) and (x_2, y_1), where y_1 (respectively, y_2) is the bridging neighbor of x_1 (respectively, x_2) in T. Moreover, the spines of $\mathcal{M}(T)$ and $\mathcal{M}'(T)$ are the same.*

By Lemma 3.8, if G has a (Δ, k)-phylogeny, then there are at least $\tau - 1$ helpful unitary (Δ, k)-QPs among $R_{2t+\tau+1}, \ldots, R_\ell$. So, if there are less than $\tau - 1$ helpful unitary (Δ, k)-QPs among $R_{2t+\tau+1}, \ldots, R_\ell$, then G has no (Δ, k)-phylogeny. In the sequel, we assume that there are at least $\tau - 1$ helpful unitary (Δ, k)-QPs among $R_{2t+\tau+1}, \ldots, R_\ell$. Without loss of generality, we may assume that $R_{2t+\tau+1}, \ldots, R_{2t+2\tau-1}$ are helpful unitary (Δ, k)-QPs.

If $\tau \geq 2$, then we connect $R_{2t+1}, \ldots, R_{2t+2\tau-1}$ into a single (Δ, k)-QP $\mathcal{R}$ as follows.

1. Introduce $\tau - 1$ bridging nodes $x_1, \ldots, x_{\tau-1}$.

2. Select a degree-2 node y_{2t+1} of R_{2t+1}, and select a degree-2 node $z_{2t+\tau}$ of $R_{2t+\tau}$.

3. For each i with $2t+2 \leq i \leq 2t+\tau-1$, select two degree-2 nodes z_i and y_i of R_i.

4. For each i with $1 \leq i \leq \tau-1$, add edges (x_i, y_{2t+i}) and (x_i, z_{2t+i+1}), and add an edge from x_i to an (arbitrarily chosen) port node of $R_{2t+\tau+i}$.

If $\tau = 1$, we let $\mathcal{R} = R_{2t+1}$.

Note that $\mathcal{R}$ is a troublesome (Δ, k)-QP. By Lemma 3.8 and Corollaries 3.9 and 3.10, if G has a (Δ, k)-phylogeny, then G has one T such that $\mathcal{R}, R_{2t+2\tau}, \ldots, R_\ell$ are subtrees of T. In the remainder of this section, a (Δ, k)-phylogeny of G always means such a tree T. Let h be the number of degree-2 nodes in $\mathcal{R}$. Let $x_1, \ldots, x_h$ be the degree-2 nodes of $\mathcal{R}$.

Lemma 3.11 *If G has a (Δ, k)-phylogeny, then it has one T such that for all but one $x_i \in \{x_1, \ldots, x_h\}$, the connected component of $T - \{x_i\}$ containing no node of $\mathcal{R}$ is a double (Δ, k)-QP.*

Proof. Let T be a (Δ, k)-phylogeny of T. Root T at a true leaf of $\mathcal{R}$. For each $i \in \{1, \ldots, h\}$, exactly one child of x_i in T is not a node of $\mathcal{R}$; let y_i be the child. Note that $y_1, \ldots, y_h$ are bridging nodes. We say that a node y_i with $1 \leq i \leq h$ is *heavy* in T if the subtree of T rooted at y_i is not a double (Δ, k)-QP. Let h' be the number of heavy nodes among $y_1, \ldots, y_h$. If $h' \leq 1$, then T is as required. So, assume that $h' \geq 2$. We may further assume that the heavy nodes are $y_1, \ldots, y_{h'}$. We can modify T by performing the following steps for $i = 2, 3, \ldots, h'$ (in this order):

1. Find a bridging descendant z_1 of y_1 in T such that $d_T(y_1, z_1) \geq d_T(y_1, z_2)$ for every bridging descendant z_2 of y_1 in T. (Comment: The subtree of T rooted at z_1 must be a double (Δ, k)-QP. So, $z_1 \neq y_1$.)

2. Let z_2 be the parent of z_1 in T. (Comment: Possibly, $z_2 = y_1$.)

3. Delete edges (x_i, y_i) and (z_2, z_1), and further add edges (x_i, z_1) and (z_2, y_i). (Comment: Before and after this step, y_i and z_1 are bridging nodes not contained in $\mathcal{R}$. So, after this step, $\mathcal{R}$ and $R_{2t+2\tau}, \ldots, R_\ell$ remain to be subtrees of T, T remains to be a (Δ, k)-phylogeny of G, and x_i is no longer heavy in T.)

Obviously, after the above modification, T is as required. □

By Lemma 3.11, if G has a (Δ, k)-phylogeny, then there are at least $2h-2$ helpful unitary (Δ, k)-QPs among $R_{2t+2\tau}, \ldots, R_\ell$. So, if there are less than $2h-2$ helpful unitary (Δ, k)-QPs among $R_{2t+2\tau}, \ldots, R_\ell$, then G has no (Δ, k)-phylogeny. In the sequel, we assume that there are at least $2h-2$ helpful unitary (Δ, k)-QPs among $R_{2t+2\tau}, \ldots, R_\ell$. We may further assume that $R_{2t+2\tau}, \ldots, R_{2t+2\tau+2h-3}$ are helpful unitary (Δ, k)-QPs. For each $i \in \{2t+2\tau, \ldots, 2t+2\tau+2h-3\}$, let z_i be an (arbitrarily chosen) port node of R_i.

We connect $\mathcal{R}, R_{2t+2\tau}, \ldots, R_{2t+2\tau+2h-3}$ into a single (helpful) (Δ, k)-QP $\mathcal{R}'$ by performing the following steps:

1. Introduce $h-1$ bridging nodes $s_1, \ldots, s_{h-1}$.

2. For each $i \in \{1, \ldots, h-1\}$, add edges $(s_i, z_{2t+2\tau+2i-2})$, $(s_i, z_{2t+2\tau+2i-1})$, and (s_i, x_i).

Now, we are left with $\mathcal{R}', R_{2t+2\tau+2h-2}, \ldots, R_\ell$ each of which is helpful or moderate. Moreover, by Lemma 3.11, if G has a (Δ, k)-phylogeny, then it has one in which $\mathcal{R}', R_{2t+2\tau+2h-2}, \ldots, R_\ell$ are subtrees. So, we can modify the proof of Lemma 3.5 to show that G has a (Δ, k)-phylogeny if and only if $a' \geq b' + 3$, where a' (respectively, b') is the number of helpful (respectively, moderate) (Δ, k)-QPs among $\mathcal{R}', R_{2t+2\tau+2h-2}, \ldots, R_\ell$.

In summary, we have the following:

Theorem 3.12 *Suppose that k is odd. Then, we can decide if G has a (Δ, k)-phylogeny, and construct one if so, in linear time.*

3.2 The Case where k is Even

Throughout this subsection, we assume that k is even. The contents in this subsection are very similar to those in the last subsection. In particular, the lemmas in this subsection one-to-one correspond to the lemmas in the last subsection. Moreover, the proof of each lemma in this subsection is very similar to (indeed a bit simpler than) its corresponding lemma in the last subsection. So, we will omit the proofs of the lemmas.

Lemma 3.13 *Suppose that each Q_i ($1 \leq i \leq \ell$) is helpful or moderate. Then, G has a (Δ, k)-phylogeny if and only if $a \geq 2$, where a is the number of helpful (Δ, k)-QPs among $Q_1, \ldots, Q_\ell$.*

In the sequel, we assume that at least one Q_i ($1 \leq i \leq \ell$) is troublesome or dangerous (since otherwise Lemma 3.13 solves the problem).

Let T be a (Δ, k)-phylogeny of G. For each dangerous Q_i ($1 \leq i \leq \ell$), we say that a false leaf x of Q_i is *active* in T, if no connected components of $T - \{x\}$ is a helpful unitary (Δ, k)-QP. A dangerous Q_i ($1 \leq i \leq \ell$) is *active* in T if at least one false leaf of Q_i is active in T.

Lemma 3.14 *Suppose that G has a (Δ, k)-phylogeny. Then, G has a (Δ, k)-phylogeny T such that no dangerous unitary (Δ, k)-QP is active in T.*

Let I be the set of all $i \in \{1, \ldots, \ell\}$ such that Q_i is dangerous. For each $i \in I$, let t_i be the number of false leaves in Q_i. Let $t = \sum_{i \in I} t_i$. By Lemma 3.14, if G has a (Δ, k)-phylogeny, then there are at least t helpful unitary (Δ, k)-QPs. So, if there are less than t helpful unitary (Δ, k)-QPs, then G has no (Δ, k)-phylogeny. In the sequel, we assume that there are at least t helpful unitary (Δ, k)-QPs. Without loss of generality, we may assume that $Q_1, \ldots, Q_t$ are helpful.

We connect $Q_1, \ldots, Q_t$ to the dangerous unitary (Δ, k)-QPs as follows.

1. Construct a one-to-one correspondence between $Q_1, \ldots, Q_t$ and the t false leaves of the dangerous unitary (Δ, k)-QPs.

2. For each $i \in \{1, \ldots, t\}$, add an edge from an (arbitrarily chosen) port node of Q_i to the false leaf corresponding to Q_i.

The above modification extends each dangerous unitary (Δ, k)-QP Q_i to a troublesome (Δ, k)-QP R_i. For convenience, let $R_i = Q_i$ for each $i \in \{t+1, \ldots, \ell\}$ such that Q_i is not dangerous.

Now, we are left with $R_{t+1}, \ldots, R_\ell$; none of them is dangerous. Let τ be the number of troublesome (Δ, k)-QPs among $R_{t+1}, \ldots, R_\ell$. Note that $\tau = |i \in \{1, \ldots, \ell\} \mid Q_i$ is troublesome or

dangerous}. So, $\tau \geq 1$. Without loss of generality, we may assume that $R_{t+1}, \ldots, R_{t+\tau}$ are troublesome.

By Lemma 3.14, if G has a (Δ, k)-phylogeny, then it has one in which $R_{t+1}, \ldots, R_\ell$ are subtrees. So, in the remainder of this section, a (Δ, k)-phylogeny of G always means one in which $R_{t+1}, \ldots, R_\ell$ are subtrees.

For each (Δ, k)-phylogeny T of G, let $\mathcal{M}(T)$ denote the tree obtained by modifying T by merging each R_i with $t+1 \leq i \leq \ell$ into a super-node. For convenience, we abuse the notation to let each R_i also denote the super-node corresponding to R_i in $\mathcal{M}(T)$.

Lemma 3.15 *If G has a (Δ, k)-phylogeny, then it has one T such that there is a path in $\mathcal{M}(T)$ on which $R_{t+1}, \ldots, R_{t+\tau}$ appear.*

Lemma 3.16 *If G has a (Δ, k)-phylogeny, then it has one T such that there is a path in $\mathcal{M}(T)$ whose nodes are exactly $R_{t+1}, \ldots, R_{t+\tau}$.*

In the remainder of this section, a (Δ, k)-phylogeny of G always means one T such that there is a path q in $\mathcal{M}(T)$ whose nodes are exactly $R_{t+1}, \ldots, R_{t+\tau}$. We call q the *spine* of $\mathcal{M}(T)$. Obviously, Corollaries 3.9 and 3.10 still hold even if k is even.

If $\tau \geq 2$, then we connect $R_{t+1}, \ldots, R_{t+\tau}$ into a single (Δ, k)-QP $\mathcal{R}$ as follows.

1. Select a degree-2 node y_{t+1} of R_{t+1}, and select a degree-2 node $z_{t+\tau}$ of $R_{t+\tau}$.
2. For each i with $t+2 \leq i \leq t+\tau-1$, select two degree-2 nodes z_i and y_i of R_i.
3. For each i with $t+1 \leq i \leq t+\tau-1$, add edge (y_i, z_{i+1}).

If $\tau = 1$, we let $\mathcal{R} = R_{t+1}$.

Note that $\mathcal{R}$ is a troublesome (Δ, k)-QP. By Lemma 3.16 and Corollaries 3.9 and 3.10, if G has a (Δ, k)-phylogeny, then G has one T such that $\mathcal{R}, R_{t+\tau+1}, \ldots, R_\ell$ are subtrees of T. In the remainder of this section, a (Δ, k)-phylogeny of G always means such a tree T. Let h be the number of degree-2 nodes in $\mathcal{R}$. Let $x_1, \ldots, x_h$ be the degree-2 nodes of $\mathcal{R}$.

Lemma 3.17 *If G has a (Δ, k)-phylogeny, then it has one T such that for all but one $x_i \in \{x_1, \ldots, x_h\}$, the connected component of $T - \{x_i\}$ containing no node of $\mathcal{R}$ is a helpful unitary (Δ, k)-QP.*

By Lemma 3.17, if G has a (Δ, k)-phylogeny, then there are at least $h-1$ helpful unitary (Δ, k)-QPs among $R_{t+\tau+1}, \ldots, R_\ell$. So, if there are less than $h-1$ helpful unitary (Δ, k)-QPs among $R_{t+\tau+1}, \ldots, R_\ell$, then G has no (Δ, k)-phylogeny. In the sequel, we assume that there are at least $h-1$ helpful unitary (Δ, k)-QPs among $R_{t+\tau+1}, \ldots, R_\ell$. We may further assume that $R_{t+\tau+1}, \ldots, R_{t+\tau+h-1}$ are helpful unitary (Δ, k)-QPs. For each $i \in \{t+\tau+1, \ldots, t+\tau+h-1\}$, let z_i be an (arbitrarily chosen) port node of R_i.

We connect $\mathcal{R}, R_{t+\tau+1}, \ldots, R_{t+\tau+h-1}$ into a single (helpful) (Δ, k)-QP $\mathcal{R}'$ by adding edges $(x_1, z_{t+\tau+1}), \ldots, (x_{h-1}, z_{t+\tau+h-1})$.

Now, we are left with $\mathcal{R}', R_{t+\tau+h}, \ldots, R_\ell$ each of which is helpful or moderate. Moreover, by Lemma 3.17, if G has a (Δ, k)-phylogeny, then it has one in which $\mathcal{R}', R_{t+\tau+h}, \ldots, R_\ell$ are subtrees. So, we can modify the proof of Lemma 3.13 to show that G has a (Δ, k)-phylogeny if and only if $a' \geq 2$, where a' is the number of helpful (Δ, k)-QPs among $\mathcal{R}', R_{t+\tau+h}, \ldots, R_\ell$.

In summary, we have the following:

Theorem 3.18 *Suppose that k is even. Then, we can decide if G has a (Δ, k)-phylogeny, and construct one if so, in linear time.*

4 Algorithm for Optimal Quasi-Phylogeny

Throughout this section, fix a connected graph $G = (V, E)$ and two integers $k \geq 4$ and $\Delta \geq 3$. If V consists of only one vertex v, then G has only one optimal (Δ, k)-QP Q, namely, Q is the (Δ, k)-QP of G in which each internal node is of degree 3 and the distance from each false leaf to v is exactly $\lfloor \frac{k}{2} \rfloor$. So, the problem is easy when $|V| = 1$. In the remainder of this section, we assume that $|V| \geq 2$.

Consider a (Δ, k)-QP Q of G. A node x of Q is *redundant*, if x is not a true leaf of Q and one connected component of $Q - \{x\}$ contains all true leaves of Q. We can obtain a new tree P by removing all redundant nodes from Q. Note that V is exactly the set of all leaves of P and P has at least one internal node of degree $\leq \Delta - 1$. We call P a *degree-Δ kth root semi-phylogeny* ((Δ, k)-SP for short) of G. We define the *cost* of P to be $\max\{1, \sum_x 2^{\lfloor k/2 \rfloor - g(x)}\}$, where x ranges over all degree-2 nodes of P and $g(x) = \min_{v \in V} d_P(x, v)$. Obviously, the following hold:

- The cost of P does not exceed the cost of Q.
- If Q is optimal, then the costs of Q and P are equal (cf. Lemma 2.1).
- Some subtree of Q is a (Δ, k)-QP of G and its cost is equal to the cost of P.

P is an *optimal* (Δ, k)-SP of G if its cost is minimized over all (Δ, k)-SPs of G. Obviously, an optimal (Δ, k)-QP of G can be computed from an optimal (Δ, k)-SP of G in linear time.

So, in the remainder of this section, we will design a linear-time algorithm for computing an optimal $(3, k)$-SP of G. The adaptation to arbitrary Δ is straightforward and is hence omitted here. The algorithm is a modification of the algorithm in [2] for computing a kth root phylogeny for a given connected graph.

4.1 Basic Definitions

A *tree-decomposition* of G is a pair $\mathcal{D} = (\mathcal{T}, \mathcal{B})$ consisting of a tree $\mathcal{T} = (\mathcal{U}, \mathcal{F})$ and a collection $\mathcal{B} = \{B_\alpha \mid B_\alpha \subseteq V, \alpha \in \mathcal{U}\}$ of *sets* (called *bags*) such that

- $\bigcup_{\alpha \in \mathcal{U}} B_\alpha = V$,
- for each edge $(v_1, v_2) \in E$, there is a node $\alpha \in \mathcal{U}$ such that $\{v_1, v_2\} \subseteq B_\alpha$, and
- if $\alpha \in \mathcal{U}$ is on the path connecting β and γ in $\mathcal{T}$, then $B_\beta \cap B_\gamma \subseteq B_\alpha$.

A useful property is that if G is connected, then for every two adjacent nodes α and β in $\mathcal{T}$, $B_\alpha \cap B_\beta \neq \emptyset$.

In the sequel, we abuse the notations to use $\mathcal{D}$ to denote the tree $\mathcal{T}$ in it, and denote the bag associated with a node α of $\mathcal{D}$ by B_α. For two adjacent nodes α and β of $\mathcal{D}$, let $U(\alpha, \beta) = \bigcup_\gamma B_\gamma$, where γ ranges over all nodes of $\mathcal{D}$ with $d_{\mathcal{D}}(\gamma, \alpha) < d_{\mathcal{D}}(\gamma, \beta)$. In other words, if we root $\mathcal{D}$ at node β, then $U(\alpha, \beta)$ is the union of the bags associated with α and its descendants in $\mathcal{D}$. A useful property of $\mathcal{D}$ is that for every internal node α and every two neighbors β and γ of α in $\mathcal{D}$, G has no edge between any vertex of $U(\beta, \alpha) - B_\alpha$ and any vertex of $U(\gamma, \alpha) - B_\alpha$.

A *clique-tree-decomposition* of G is a tree-decomposition $\mathcal{D}$ of G such that for each node α in $\mathcal{D}$, B_α is a maximal clique of G. (Recall that a clique of G is *maximal* if it is not contained in a larger clique of G.)

As pointed out in [2], if G has a (Δ, k)-phylogeny, then we can compute a clique-tree-decomposition $\mathcal{D}$ of G in linear time that satisfies the following two conditions:

1. For each node α in $\mathcal{D}$, $|B_\alpha| \leq 3 \cdot 2^{(k-2)/2}$ if k is even, while $|B_\alpha| \leq 2^{(k+1)/2}$ if k is odd.
2. The degree of each node α in $\mathcal{D}$ is exactly 3.

An optimal (Δ, k)-SP of G is computed by a dynamic programming algorithm applied on a rooted version of the decomposition tree $\mathcal{D}$. The details of the dynamic programming algorithm are developed in the next two subsections.

4.2 Ideas behind the Dynamic Programming Algorithm

Note that every internal node in a $(3, k)$-SP T of G has degree at most 3.

Definition 4.1 *Let W be a nonempty set of vertices of G. A* relaxed semi-phylogeny *(RSP for short) for W is a tree R satisfying the following four conditions:*

- *The degree of each node in R is at most* 3.
- *Each vertex of W is a leaf in R and appears in R only once. For convenience, we call the leaves of R that are also vertices of W* final leaves *of R, and call the other leaves of R* temporary leaves *of R.*
- *For every two vertices u and v of W, u and v are adjacent in G if and only if $d_R(u, v) \leq k$.*
- *Each temporary leaf x of R is assigned a pair (γ, t), where γ is a node of $\mathcal{D}$ and t is a nonnegative integer. We call γ the* color *of x and call t the* threshold *of x. For convenience, we denote the color of a temporary leaf x of R by $c_R(x)$ and denote the threshold of x by $t_R(x)$.*

By $L_t(R)$, we denote the set of temporary leaves in R. For each degree-2 node x in R, let $\lambda_R(x) = \min_{u \in W} d_R(x, u)$ if $L_t(R) = \emptyset$, while let $\lambda_R(x) = \min\{\min_{u \in W} d_R(x, u), \min_{y \in L_t(R)} d_R(x, y) + t_R(y)\}$ if $L_t(R) \neq \emptyset$. The cost *of R, denoted by $cost(R)$, is $\sum_x 2^{\lfloor k/2 \rfloor - \lambda_R(x)}$, where x ranges over all degree-2 nodes in R.*

Note that if $W = V$ and R has at least one degree-2 node but has no temporary leaf, then R is a $(3, k)$-SP of G.

Intuitively speaking, the temporary leaves of R serve as ports from which we can expand R so that it may eventually become a $(3, k)$-SP R' of G. In more detail, for each temporary leaf x of R, $c_R(x)$ specifies the node whose bag contains the leaf $v \in V - W$ of R' closest to x in R' and $t_R(x)$ specifies the distance between x and v in R'.

Our algorithm processes the nodes of $\mathcal{D}$ one by one. While processing a node α of $\mathcal{D}$, the algorithm finds out all RSPs for B_α that are possibly subtrees of $(3, k)$-SPs of graph G. The following lemma shows that such RSPs for B_α have certain useful properties.

Lemma 4.1 *Let T be a $(3,k)$-SP of G. Let α be a node of $\mathcal{D}$. Root T at an arbitrary leaf that is in B_α. Define a* pure *node to be a node x of T such that α has a neighbor γ in $\mathcal{D}$ such that all leaf descendants of x in T are in $U(\gamma,\alpha) - B_\alpha$. Define a* critical *node to be a pure node of T whose parent is not pure. Let R be the RSP for B_α obtained from T by performing the following steps:*

1. *For every critical node x of T, perform the following:*
   (a) *Find the neighbor γ of α such that all leaf descendants of x in T are contained in $U(\gamma,\alpha)$.*
   (b) *Compute the minimum distance from x to a leaf descendant of x in T; let i_x denote this distance. (*Comment: *$i_x \leq k$ or else the leaf descendants of x in tree T would be unreachable from the outside in graph G.)*
   (c) *Delete all descendants (excluding x) of x.*
   (d) *Let x be a temporary leaf, and assign the pair (γ, i_x) to x.*
2. *Unroot T.*

Then, the resultant tree R has the following properties:

- *For every temporary leaf x of R, $c_R(x)$ is a neighbor of α in $\mathcal{D}$ and $0 \leq t_R(x) \leq k$.*
- *For every two temporary leaves x and y of R with different colors, it holds that $t_R(x)+t_R(y)+d_R(x,y) > k$.*
- *For every neighbor γ of α in $\mathcal{D}$, every temporary leaf x of R with $c_R(x) = \gamma$, and every final leaf u of R with $u \notin B_\gamma$, it holds that $d_R(x,u) + t_R(x) > k$.*
- *For every internal node x of R, either there is a final leaf u of R with $d_R(u,x) \leq k-1$, or at least one descendant of x in R' is a final leaf of R where R' is obtained from R by rooting it at an arbitrary final leaf.*

Proof. Same as that of Lemma 3.2 in [2]. □

Each RSP R for B_α having the four properties in Lemma 4.1 is called a *skeleton* of α. The following lemma shows that there can be only a constant number of skeletons of α.

Lemma 4.2 *For each node α of $\mathcal{D}$, the number of skeletons of α is bounded from above by a constant depending only on k and $|B_\alpha|$.*

Proof. Same as that of Lemma 3.3 in [2]. □

Definition 4.2 *Let α and β be two adjacent nodes of $\mathcal{D}$. Let S be a skeleton of α. An* expansion *of S to $U(\alpha,\beta)$ is an RSP X for $U(\alpha,\beta)$ such that some subtree Y of X is isomorphic to S, and the bijection f from the node set of S to the node set of Y witnessing this isomorphism satisfies the following conditions:*

- *For every final leaf v of S, $f(v) = v$.*
- *For every temporary leaf y of S with $c_S(y) = \beta$, $f(y)$ is a temporary leaf of X with $c_X(f(y)) = c_S(y)$ and $t_X(f(y)) = t_S(y)$.*

- *Suppose that we root X at a leaf $r \in B_\alpha \cap B_\beta$. Then, for every temporary leaf y of P with $c_S(y) \neq \beta$, all leaf descendants of $f(y)$ in X are final leaves and are contained in $U(\alpha, \beta) - B_\alpha$, the minimum distance between $f(y)$ and a leaf descendant of $f(y)$ in X is equal to $t_S(y)$.*

An expansion of S to $U(\alpha, \beta)$ is perfect *if it has no degree-2 node; otherwise, it is* imperfect. *An imperfect expansion of S to $U(\alpha, \beta)$ is* optimal *if its cost is minimized over all imperfect expansions of S to $U(\alpha, \beta)$.*

Note that a skeleton S of α may have no expansion to $U(\alpha, \beta)$. However, if G has a $(3, k)$-SP T, then the skeleton S of α obtained from T as in Lemma 4.1 has an expansion to $U(\alpha, \beta)$; more specifically, the restriction of T to $U(\alpha, \beta)$ (defined below) is an expansion of S to $U(\alpha, \beta)$.

Definition 4.3 *Let α and β be two adjacent nodes in $\mathcal{D}$. Let X be an RSP for a superset of $U(\alpha, \beta)$ such that for every temporary leaf y of X, $d_X(\beta, c_X(y)) < d_X(\alpha, c_X(y))$. The* restriction *of X to $U(\alpha, \beta)$ is an RSP for $U(\alpha, \beta)$ obtained from X by performing the following steps:*

1. *Change each final leaf $v \notin U(\alpha, \beta)$ to a temporary leaf; set the threshold of v to be 0 and set the color of v to be β.*
2. *Root X at an arbitrary vertex of $B_\alpha \cap B_\beta$.*
3. *Find those nodes x in X such that (i) every leaf descendant of x in X is a temporary leaf whose color is not α, but (ii) the parent of x in X does not have property (i).*
4. *For each node x found in the last step, if x is a leaf in X then set the color of x to be β; otherwise, perform the following steps:*
   (a) *Set $i_x = \min_y\{t_X(y) + d_X(x, y)\}$ where y ranges over all leaf descendants of x in X.*
   (b) *Delete all descendants (excluding x) of x in X.*
   (c) *Let x be a temporary leaf, and assign the pair (β, i_x) to x.*
5. *Unroot X.*

By Lemma 4.2, while processing a node α of $\mathcal{D}$, our algorithm can find out all skeletons of α in constant time. For each skeleton S of α, the algorithm tries to compute both a perfect expansion X_S and an optimal imperfect expansion $\overline{X_S}$ of S to $U(\alpha, \beta)$, where β is the parent of α in rooted $\mathcal{D}$. If X_S (respectively, $\overline{X_S}$) does not exist, the algorithm sets $X_S = \emptyset$ (respectively, $\overline{X_S} = \emptyset$). The algorithm records $(S, X_S, \overline{X_S})$ in the dynamic programming table for later use when processing the parent β. The following definition aims at removing unnecessary skeletons of α from the dynamic programming table.

Definition 4.4 *Let α and β be two adjacent nodes of $\mathcal{D}$. Let X be a skeleton of β. The* projection *of X to α is an RSP for $B_\alpha \cap B_\beta$ obtained from X by performing the five steps in Definition 4.3.*

Obviously, two different skeletons of α may have the same projection to β. For convenience, we say that these skeletons are *equivalent*. Among equivalent skeletons of α, our algorithm will record only the one whose optimal expansion to $U(\alpha, \beta)$ has the minimum cost, in the dynamic programming table. This motivates the following definition:

Definition 4.5 *Let α and β be two adjacent nodes of $\mathcal{D}$. For each skeleton S of α, the projection P of S to β is called a* projection *of α to β, and each expansion of S to $U(\alpha, \beta)$ is called an* expansion *of P to $U(\alpha, \beta)$.*

Let P be a projection of α to β. An expansion of P to $U(\alpha, \beta)$ is perfect *if it has no degree-2 node; otherwise, it is* imperfect*. An imperfect expansion of P to $U(\alpha, \beta)$ is* optimal *if its cost is minimized over all imperfect expansions of P to $U(\alpha, \beta)$.*

As a skeleton of α may have no expansion to $U(\alpha, \beta)$, a projection of α to β may have no expansion to $U(\alpha, \beta)$. However, if G has a $(3, k)$-SP T, then for the skeleton S of α obtained from T as in Lemma 4.1, the projection P of S to β has an expansion to $U(\alpha, \beta)$; more specifically, the restriction of T to $U(\alpha, \beta)$ is an expansion of P to $U(\alpha, \beta)$.

Recall that while processing a node α of $\mathcal{D}$, our algorithm tries to compute, for every skeleton S of α, both a perfect expansion X_S and an optimal imperfect expansion $\overline{X_S}$ of S to $U(\alpha, \beta)$ where β is the parent of α in rooted $\mathcal{D}$. The following lemmas show that the computation of X_S and $\overline{X_S}$ only needs both a perfect expansion and an optimal imperfect expansion of P_γ to $U(\gamma, \alpha)$ for every child γ of α in rooted $\mathcal{D}$, where P_γ is the projection of S to γ.

Lemma 4.3 *Let α be an internal node of $\mathcal{D}$. Let β, γ_1, and γ_2 be the neighbors of α in $\mathcal{D}$. Let S be a skeleton of α. For each $i \in \{1, 2\}$, let P_i be the projection of S to γ_i, and let R_i be an expansion of P_i to $U(\gamma_i, \alpha)$. Then, there is an expansion X of S to $U(\alpha, \beta)$ such that*

$$cost(X) = cost(S) + cost(R_1) + cost(R_2) - \sum_{y \in \Psi_1} 2^{\lfloor \frac{k}{2} \rfloor - \lambda_{P_1}(y)} - \sum_{y \in \Psi_2} 2^{\lfloor \frac{k}{2} \rfloor - \lambda_{P_2}(y)},$$

where Ψ_i ($i = 1$, 2) is the set of degree-2 nodes in P_i. Moreover, X is perfect if and only if S has no degree-2 node and both R_1 and R_2 are perfect.

Proof. By definition, we may view P_i as a subtree of both R_i and S for each $i \in \{1, 2\}$. By combining S, R_1, and R_2, we can obtain an expansion X of S to $U(\alpha, \beta)$ as follows:

1. For each $i \in \{1, 2\}$, delete those nodes x from R_i such that x is a node of P_i but is not a temporary leaf of P_i with $c_{P_i}(x) = \gamma_i$.
2. For each $i \in \{1, 2\}$ and for each temporary leaf x of S with $c_S(x) = \gamma_i$, modify S by replacing x with the connected component (a tree) of R_i in which x appears.

Obviously, S, P_1, P_2, R_1, and R_2 are subtrees of X. For two trees Y and Z among S, P_1, P_2, R_1, and R_2, let $Y \cap Z$ denote the subtree shared by Y and Z. Then, we have the following simple observations:

- $R_1 \cap S = P_1$, $R_2 \cap S = P_2$, and $R_1 \cap R_2 = P_1 \cap P_2$.
- For each degree-2 node y of S, $\lambda_S(y) = \lambda_X(y)$ (cf. Definition 4.1).
- For each $i \in \{1, 2\}$ and for each degree-2 node y of R_i, $\lambda_{R_i}(y) = \lambda_X(y)$.
- For each $i \in \{1, 2\}$ and for each degree-2 node y of P_i, $\lambda_{P_i}(y) = \lambda_{R_i}(y) = \lambda_X(y)$.

By the above observations, the equality in the lemma holds. The second assertion in the lemma is obvious, too. □

Lemma 4.4 *Let $\alpha, \beta, \gamma_1, \gamma_2, S, P_1, P_2, \Psi_1, \Psi_2$ be as in Lemma 4.3. The following hold:*

- *Suppose that X is a perfect expansion of S to $U(\alpha, \beta)$. Then, for each $i \in \{1, 2\}$, the restriction R_i of X to $U(\gamma_i, \alpha)$ is a perfect expansion of P_i to $U(\gamma_i, \alpha)$.*
- *Suppose that X is an optimal imperfect expansion of S to $U(\alpha, \beta)$. Then, for each $i \in \{1, 2\}$, the restriction R_i of X to $U(\gamma_i, \alpha)$ is either a perfect expansion or an optimal imperfect expansion of P_i to $U(\gamma_i, \alpha)$.*

PROOF. For each $i \in \{1, 2\}$, R_i is clearly an expansion of P_i to $U(\gamma_i, \alpha)$. Moreover, by combining S, R_1, and R_2, we can obtain X as in the proof of Lemma 4.3. Thus, we have the equality in Lemma 4.3. In turn, by the optimality of X, both R_1 and R_2 are optimal. □

4.3 Details of the Dynamic Programming Algorithm

To compute an optimal $(3, k)$-SP of G, we perform a dynamic programming on the tree-decomposition $\mathcal{D}$ as follows. To simplify the description of the algorithm, we add a new node r to $\mathcal{D}$, connect r to an arbitrary leaf α of $\mathcal{D}$, and copy the bag at α to r (that is, $B_r = B_\alpha$). Clearly, the resultant $\mathcal{D}$ is still a required tree-decomposition of G. Root $\mathcal{D}$ at r.

The dynamic programming starts at the leaves of $\mathcal{D}$, and proceeds upwards. After the unique child of the root r of $\mathcal{D}$ is processed, we will know whether G has a $(3, k)$-SP or not, and will get an optimal $(3, k)$-SP of G if any. The invariant maintained during the dynamic programming is that after each nonroot node α has been processed, for each projection P of α to its parent β, we will have obtained and stored the quintuple $(P, X_P, b_P, \overline{X_P}, \overline{b_P})$ in the dynamic programming table, where $X_P = \emptyset$ and $b_P = +\infty$ (respectively, $\overline{X_P} = \emptyset$ and $\overline{b_P} = +\infty$) if P has no perfect (respectively, imperfect) expansion to $U(\alpha, \beta)$, while X_P (respectively, $\overline{X_P}$) is a perfect expansion (respectively, an optimal imperfect expansion) of P to $U(\alpha, \beta)$ and $b_P = cost(X_P) = 0$ (respectively, $\overline{b_P} = cost(\overline{X_P})$) otherwise.

Now consider how a nonroot node α of $\mathcal{D}$ is processed. Let β be the parent of α in $\mathcal{D}$. First suppose that α is a leaf of $\mathcal{D}$. For each projection P of α to β, a perfect (respectively, imperfect) expansion of P to $U(\alpha, \beta)$ is a skeleton S of α such that P is the projection of S to β and S has no (respectively, at least one) degree-2 node. Thus, it is easy to process α.

Next suppose that α is neither a leaf nor the root node of $\mathcal{D}$, and suppose that all descendants of α in $\mathcal{D}$ have been processed. Let γ_1 and γ_2 be the children of α in $\mathcal{D}$. To process α, we first compute all projections P of α to β, and initialize $(P, X_P, b_P, \overline{X_P}, \overline{b_P}) = (P, \emptyset, +\infty, \emptyset, +\infty)$. Then, we try all skeletons S of α. When trying S, we perform the following steps for S:

1. For each $i \in \{1, 2\}$, compute the projection P_i of S to γ_i, and search the dynamic programming table to find the quintuple $(P_i, X_{P_i}, b_{P_i}, \overline{X_{P_i}}, \overline{b_{P_i}})$.
2. Compute the projection P of S to β.
3. For each $R_1 \in \{X_{P_1}, \overline{X_{P_1}}\}$ and each $R_2 \in \{X_{P_2}, \overline{X_{P_2}}\}$ such that $R_1 \neq \emptyset$ and $R_2 \neq \emptyset$, perform the following steps:

(a) Obtain an expansion X of S to $U(\alpha, \beta)$ by combining S, R_1, and R_2 as in the proof of Lemma 4.3.

(b) If S has no degree-2 node, $R_1 = X_{P_1}$, $R_2 = X_{P_2}$, and $cost(X) < b_P$, then update the quintuple $(P, X_P, b_P, \overline{X_P}, \overline{b_P})$ by replacing X_P and b_P with X and $cost(X)$ $(= 0)$, respectively.

(c) If S has a degree-2 node or $R_1 = \overline{X_{P_1}}$ or $R_2 = \overline{X_{P_2}}$, and $cost(X) < \overline{b_P}$, then update the quintuple $(P, X_P, b_P, \overline{X_P}, \overline{b_P})$ by replacing $\overline{X_P}$ and $\overline{b_P}$ with X and $cost(X)$, respectively.

Finally, let α be the unique child of the root r of $\mathcal{D}$ and suppose that α has been processed. By searching the dynamic programming table, we try to find all projections P of α to r such that (i) P has no temporary leaf whose color is r, and (ii) the quintuple $(P, X_P, b_P, \overline{X_P}, \overline{b_P})$ satisfies that $\overline{X_P} \neq \emptyset$. If no such projection P is found, then we can conclude that G has no $(3, k)$-SP; otherwise, among the expansions $\overline{X_P}$ found, the one with the minimum cost is an optimal $(3, k)$-SP of G.

The above discussion justifies the following theorem:

Theorem 4.5 *Let $k \geq 4$ be an integral constant. There is a linear-time algorithm determining if a given connected graph has a $(3, k)$-SP, and if so, demonstrating an optimal $(3, k)$-SP.*

Theorem 4.5 can be easily generalized to arbitrary $\Delta \geq 3$. Thus, Lemma 3.1 follows immediately.

References

[1] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: a Survey*, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[2] Z.-Z. Chen, T. Jiang, and G.-H. Lin, *Computing phylogenetic roots with bounded degrees and errors*, SIAM Journal on Computing, 32 (2003) 864–879.

[3] P. E. Kearney and D. G. Corneil, *Tree powers*, Journal of Algorithms, 29 (1998) 111–131.

[4] G.-H. Lin, P. E. Kearney, and T. Jiang, *Phylogenetic k-root and Steiner k-root*, in: The 11th Annual International Symposium on Algorithms and Computation (ISAAC 2000), Lecture Notes in Computer Science, 1969 (2000) 539–551.

[5] Y.-L. Lin and S. S. Skiena, *Algorithms for square roots of graphs*, SIAM Journal on Discrete Mathematics, 8 (1995) 99–118.

[6] R. Motwani and M. Sudan, *Computing roots of graphs is hard*, Discrete Applied Mathematics, 54 (1994) 81–88.

[7] N. Nishimura, P. Ragde, and D. M. Thilikos, *On graph powers for leaf-labeled trees*, in: Proceedings of the 7th Scandinavian Workshop on Algorithm Theory (SWAT 2000), Lecture Notes in Computer Science, 1851 (2000) 125–138.

[8] D. L. Swofford, G. J. Olsen, P. J. Waddell, and D. M. Hillis, *Phylogenetic inference*, in: D. M. Hillis, C. Moritz, and B. K. Mable (Ed.), Molecular Systematics (2nd Edition), Sinauer Associates, Sunderland, Massachusetts, 1996, pp. 407–514.