Figure 4. By using the Eulerian cycle \(v_1v_2v_4v_2v_3v_4v_1\) in graph \(G\), this map produces the spanning tree oriented toward \(v_1\) indicated in bold in the graph on the right.

Since here we are claiming that \(T \in T_1\), we must prove that a) \(T\) is indeed a tree, and that b) \(T\) is oriented towards the vertex \(v_1\).

To show (a) we first assume that \(T\) contains a cycle \(C\), which would not contain \(v_1\). Also, since \(C\) is a cycle, each vertex in \(C\) has an outdegree of 1. Assume that edge \(e_g\) is the last edge of Euler trail \(S\) on cycle \(C\), which connects the vertices \(v_g\) and \(v_h\). If this is the case, then \(S\) is arriving back to \(v_h\) despite having left it for the last time after traveling through edge \(eh\) previously. This contradicts the corresponding map, which states that an edge \(e_i\) follows the last visit to vertex \(v_j\) and will not return to \(v_j\) again. Therefore, \(T\) is a tree.

Now, we prove (b) by asking whether or not \(T\) is oriented towards \(v_1\). We answer this by first assuming that \(T\) contains a path \(v_kv_{k-1}...v_1\). The edge between vertices \(v_2\) and \(v_1\) is \(e_2\), as there is no \(e_1\). Further, the edge between vertices \(v_3\) and \(v_2\) could be either \(e_2\) or \(e_3\), but since \(e_2\) has already been assigned, it must be \(e_3\). As we proceed in this same manner between all vertices in the path, we can prove that the path \(v_kv_{k-1}...v_1\) is indeed oriented towards the vertex \(v_1\).

Thus, the digraph \(T\) is in the set of spanning trees oriented towards vertex \(v_1\) in the original graph \(G\). Therefore, we can conclude that \(T \in T_1\) and continue with this proof.
To get the desired map $f_1: E_1 \rightarrow T_1$, we must set $f_1(S) = T$. Now, working backwards, the set $f_1^{-1}(T)$ describes all the Euler trails about vertex $v_1$ that give rise to the spanning tree $T$.

Thus, using the fact that $f_1(S) = T$ and $S \in E_1$, we can draw an Euler trail $S$ in the following manner. We begin at an edge which leaves vertex $v_1$. Once we return to $v_1$, we leave it through another unused edge, which also must exit $v_1$; when there are no untouched edges exiting $v_1$, the trail ends. Now, upon arriving at $v_j$ (where $j > 1$), leave this vertex by an edge that is not $e_j$ and also not in the tree $T$; once these edges have been exhausted, then leave $v_j$ through the edge $e_j$.

Based on the fact that the indegree and outdegrees of any vertex $v_j$ in $G$ are equivalent, we have successfully found an Euler trail $S$ which is in the set $E_1$ such that $f_1(S) = T$. Note that, for some spanning trees $T$, there will be more than one Euler trail produced with a given spanning tree $T$.

Using the graph $G$ as an example, we show the process of using the spanning tree $T$ (as shown darkened in Figure 3) to construct the Euler trails $S$ that gives rise with it. Start at the edge which leaves $v_1$ towards $v_2$. Once at $v_2$, we must use the edge towards $v_3$, as the one towards $v_3$ is the “$e_j$” edge used in the tree $T$. Notice at $v_4$, there are two outgoing edges, since the one towards $v_1$ is also part of $T$, so we use the edge going back to $v_2$. Having returned to $v_2$, the only remaining edge is the one from $T$, so we use that. The same follows for $v_4$, as the only remaining edge, also from $T$, completes the Eulerian trail $S$.

The primary concept in this process is as follows: once we enter any vertex in the graph, we exit that vertex by a non-tree edge for as long as possible, until we are only left with a tree edge. Continuing to do so for each vertex produces an Eulerian trail in the graph.

In general, we can express the number of Euler trails $S$ that can be produced with a given spanning tree $T$ oriented towards $v_1$ (denoted as $|f_1^{-1}(T)|$) as follows:

$$|f_1^{-1}(T)| = d^+(v_1) \cdot \prod_{j=2}^{n} (d^+(v_j) - 1)!.$$ 

The factorial of the outdegree of vertex $v_1$ considers number of different ways the Euler trail can begin (which explains why the size of $T_1$ is greater than $E$), and the product factorial considers that there is one less available outgoing edge at vertex $v_j$ as the Euler trail $S$ passes through.

Furthermore, upon focusing on the entire set of Eulerian circuits $E$ and disregarding a set vertex $v_1$ to start from, the number of said circuits is as follows:

$$|E| = |T_1| \cdot \prod_{j=1}^{n} (d^+(v_j) - 1)!.$$ 

Here, the product factorial for outdegrees of all vertices in $G$ is multiplied by the total number of spanning trees $T_1$ oriented towards the vertex $v_1$.

With this series of equations, we have thus proven and verified the accuracy of the formula which is associated with the BEST theorem.

The BEST theorem relies on the number of spanning trees which exist in a particular Eulerian digraph. We now need a theorem which enables us to determine the number of spanning trees in a directed Eulerian graph. This theorem has been referred to as the matrix tree theorem, and a full understanding of its proof requires some further definitions. The following terms and definitions follow from texts by Fleischner [15-16], Bollobas [14], and Pezvzer [9].

Given a directed Eulerian graph $G$, it is possible to construct various matrices which encode information about the properties of the graph. One matrix, the adjacency matrix, lists the number of edges between various vertices in the graph. It is an $n \times n$ matrix where $n$ is the total number of vertices in $G$. If $a_{ij}$ is the number of directed edges that leave a vertex $v_j$ and enter a vertex $v_i$ in $G$, then $a_{ij}$ is the $i,j$-th entry of the matrix. The $i,i$-th entries, which are the diagonal of the matrix, will be zero unless $G$ has loops.

Recall the graph $G$ from Figure 3 that we just used previous. Its adjacency matrix is denoted here as $A(G)$. 

\[
\begin{bmatrix}
|f_1^{-1}(T)| = d^+(v_1) \cdot \prod_{j=2}^{n} (d^+(v_j) - 1)! \\
|E| = |T_1| \cdot \prod_{j=1}^{n} (d^+(v_j) - 1)!
\end{bmatrix}
\]
\[ A(G) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}. \]

Notice the following: a) \( \nu_1 \) sends an edge to \( \nu_2 \), hence a 1 in the 1,2-entry; b) \( \nu_2 \) sends edges to \( \nu_3 \) and \( \nu_4 \), hence 1's in the 2,3 and 2,4 entries; c) \( \nu_3 \) sends an edge to \( \nu_4 \), hence a 1 in the 3,4-entry; and d) \( \nu_4 \) sends edges to \( \nu_1 \) and \( \nu_2 \), hence 1's in the 4,1 and 4,2 entries. Also, since there are no multiple edges of the same direction in \( G \), there are only 1's in the entries. Further, the diagonal contains all zeros because there are no loops in \( G \).

A second matrix that can be constructed from a graph is the diagonal matrix. This matrix gives the indegrees of each vertex in the graph along its diagonal. All other entries must be zero. Using the same graph \( G \), its diagonal matrix \( D(G) \) is as shown:

\[ D(G) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}. \]

A third matrix combines both aspects of these two matrices \( A(G) \) and \( D(G) \) that we have described above. Some graph theorists will refer to this as the Kirchoff matrix, as seen in Fleischner [16], while others will call it the combinatorial Laplacian matrix, as seen in Bollobás [14]. For the sake of this paper, we will call it the Laplacian matrix, and denote it \( L(G) \). Specifically, this matrix is the result of subtracting \( A(G) \) from \( D(G) \).

Once again, using our graph \( G \) from Figure 3 as an example, the Laplacian matrix \( L(G) \) is as shown below:

\[ L(G) = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 1 & -1 \\ -1 & -1 & 0 & 2 \end{pmatrix}. \]

This matrix is central to the matrix tree theorem. There are certain characteristics that all Laplacian matrices of Eulerian digraphs possess. First, the diagonal still represents the indegree of each vertex in \( G \), having only removed any loops. In our example there are none, thus the diagonal stays the same. Also, for each edge that exists from vertex \( \nu_i \) to \( \nu_j \), there is a -1 in the \( i,j \)-th entry of the matrix, where \( i \) and \( j \) are not equal.

Another characteristic of the Laplacian matrix is that columns and rows sum to zero. The reason for this is clear. Each column \( i \) accounts for two quantities. One of which, on the diagonal, is the indegree of vertex \( \nu_i \) excluding loops. The second is the number of edges that are received by \( \nu_i \) from all other vertices in \( G \), whose values here are negative. Thus, it is reasonable that by adding these together, all edges that enter vertex \( \nu_i \) are being accounted for, which produces zero for the balanced digraphs we consider here.

On the other hand, each row \( i \) accounts for two different quantities. The first is the number of edges that leave vertex \( \nu_i \) excluding loops, and the second is the number of edges that are received by all other vertices from \( \nu_i \). These values, however, are all negative. Since we know \( G \) is an Eulerian digraph, the number of edges entering and number of edges exiting any vertex are equivalent. Because of this, we know that by adding the indegree of vertex \( \nu_i \) which is positive, to the summation of the edges which leave it, which is negative, will always result in zero for any \( L(G) \) where \( G \) is an Eulerian digraph.

We now recall the cofactor of a matrix. This value is found by removing the \( i \)-th row and \( i \)-th column of a given matrix, and then taking the determinant of the remaining matrix. We denote this as follows, where \( k \) is the \( i \)-th cofactor:

\[ k = \det_{i,i} L(G). \]

We have included the \( L(G) \) matrix in the formula above because, for the sake of this paper and ensuing theorem and proof, we will be considering the cofactors of the Laplacian matrices of various directed Eulerian graphs.
As an example, consider the Laplacian matrix for the graph $G$ from Figure 3. The cofactor at the first row and first column is the determinant of the matrix below:

$$
\begin{pmatrix}
2 & -1 & -1 \\
0 & 1 & -1 \\
1 & 0 & 2
\end{pmatrix}.
$$

Taking the determinant of this matrix produces the value of $k = 2$. Further, if we found the other three cofactors from $L(G)$, we would find all of them to be 2; the reason for this stems back to the fact that all rows and columns sum to zero, as this will hold true for all such matrices. Proof of this can be found in Fleischner [16]. It is important to note that matrices whose rows and columns do not add to zero may have different cofactors depending on which rows and columns are removed from the original matrix. This information allows us to move on to a theorem which is pivotal for full understanding of the former BEST theorem and other topics involving Eulerian graphs. As stated earlier, this theorem is referred to as the matrix tree theorem.

Theorem 5.2. Given a directed graph $G$ with the set of vertices $V(G) = \{v_1, \ldots, v_n\}$ and a set of spanning trees $t_i(G)$ oriented towards the vertex $v_i$, then $|t_i(G)|$ is equal to the cofactor of $L(G)$ on the $i$-th row and $i$-th column. That is,

$$|t_i(G)| = \det_{ii} L(G).$$

The following proof for the matrix tree theorem was constructed with the aid of Bollobás [14], which proves a similar theorem involving undirected and weighted graphs for electrical networks.

Proof: We will use induction on the number of edges $m$ in a graph for this proof. Further, we will assume that the given graph $G$ is connected, as an unconnected graph has zero spanning trees, and that $m$ is greater than one, as a graph with one edge only has one spanning tree.

Now, given two vertices $v_1$ and $v_2$ that are adjacent to each other in $G$, we will produce two new graphs from $G$ which have less than $m$ edges. The first is created by removing all the directed edges that leave vertex $v_1$ and enter $v_2$ in $G$, whose set we express as $v_1v_2$. This produces the graph $G_{v_1v_2}$. To make the second graph we contract these edges $v_1v_2$, producing a graph $G/v_1v_2$. In this case, the vertices $v_1$ and $v_2$ are fused, creating a new vertex $v_{12}$. Also, the edges previously coming into each former vertex from a particular vertex $v_i$ are joined as one edge, with a weight totaling the summation of these individual edges. Further, all edges exiting each removed vertex into $v_i$ are joined and weighted in the same manner. Note that both graphs $G_{v_1v_2}$ and $G/v_1v_2$ will have less than $m$ edges.

Figure 5 depicts an example of producing these two new graphs from the