Computer Science 331

Analysis of Prim's Algorithm: Proof of a Key Lemma

1 Introduction

Consider the algorithm to find a minimum-cost spanning tree that was presented and analyzed during the last full week of classes. Recall that if $\mathtt{G} = (\mathtt{V}, \mathtt{E})$ is a a weighted undirected graph that is given as input and $\mathtt{G}$ is connected then, following an initialization step, this algorithm uses a priority queue to choose vertices and edges that should be added to a subgraph

$$\mathtt{G_b} = (\mathtt{V_b}, \mathtt{E_b})$$

of the input graph $\mathtt{G}$. As documented in the lecture notes on the analysis of this algorithm, it is reasonably easy to establish that each of the following properties is satisfied after both the beginning and end of each execution of the body of the `while` loop that makes up the second part of this algorithm:

- $\mathtt{G_b}$ is a *tree*.
- $\mathtt{G_b}$ is a subgraph of some *spanning tree* $\mathtt{T}$ of $\mathtt{G}$.

The objective of this handout is to provide a proof of the next major property (that is not quite so easy to establish).

- $\mathtt{G_b}$ is a subgraph of some *minimum-cost spanning tree* $\mathtt{T}$ of $\mathtt{G}$.

The *proof* of this claim will proceed by induction on the number of executions of the body of the `while` loop that have taken place, so far.

2 Basis

One can see, by an examination of the code, that $\mathtt{V_b} = \emptyset$ and $\mathtt{E_b} = \emptyset$ immediately before the first execution of the body of the `while` loop, while $\mathtt{V_b} = \{\mathtt{s}\}$ and $\mathtt{E_b} = \emptyset$ immediately after this execution of the loop body for some vertex $\mathtt{s} \in \mathtt{V}$.

In each case, since $\mathtt{G_b} = (\mathtt{V_b}, \mathtt{E_b})$ does not include any edges at all, it is clear that $\mathtt{G_b}$ is a subgraph of *every* spanning tree of $\mathtt{G_b}$. In particular, $\mathtt{G_b}$ is a subgraph of every *minimum-cost spanning tree* of $\mathtt{G}$ and (since there are only finitely many spanning trees, so that one must have minimal cost), at least one minimum-cost spanning tree $\mathtt{T}$ of $\mathtt{G}$ does exist.

3 Inductive Step

3.1 Notation

Suppose now that $k \geq 1$ and that the subgraph $\mathtt{G_b} = (\mathtt{V_b}, \mathtt{E_b})$ produced after k executions of the body of the `while` loop is a subgraph of some minimum-cost spanning tree $\mathtt{T}$ of $\mathtt{G}$.

Suppose, as well, that the body of the loop is executed (at least) one more time. Then, of course, $\mathtt{G_b}$ is also the subgraph that has been produced at the *beginning* of the $k+1^{\text{st}}$ execution of the loop body. Let

$$\mathtt{G}'_\mathtt{b} = (\mathtt{V}'_\mathtt{b}, \mathtt{E}'_\mathtt{b})$$

be the graph that is produced *after* the $k+1^{\text{st}}$ execution of the loop body.

3.2 Proof that the New Graph is a Subgraph

In this case, one can see by inspection of the code that

$$\mathtt{V}'_\mathtt{b} = \mathtt{V_b} \cup \{\mathtt{v}\} \tag{1}$$

for some vertex $\mathtt{v}$ such that $\mathtt{v} \notin \mathtt{V_b}$, and

$$\mathtt{E}'_\mathtt{b} = \mathtt{E_b} \cup \{(\mathtt{u}, \mathtt{v})\} \tag{2}$$

for some vertex $\mathtt{u} \in \mathtt{V}$. In particular, $\mathtt{u} = \pi(\mathtt{v})$. As noted during the discussion of this algorithm in class, one can show (without too much trouble) that $(\pi(\mathtt{v}), \mathtt{v}) \in \mathtt{E}$ and $\pi(\mathtt{v})$ is `black` at this point in the computation, so that $\pi(\mathtt{v}) \in \mathtt{V_b}$ and $\mathtt{G}'_\mathtt{b}$ is a subgraph of $\mathtt{G}$.

3.3 Proof that the New Graph is a Subgraph of a Minimum-Cost Spanning Tree

In order to complete this proof it is necessary and sufficient to show that there exists a minimum-cost spanning tree $\mathtt{T}'$ of $\mathtt{G}$ such that $\mathtt{G}'_\mathtt{b}$ is a subgraph of $\mathtt{T}'$.

Two cases can be identified and should be considered: Either the edge $(\mathtt{u}, \mathtt{v})$ is an edge in $\mathtt{T}$ or it is not.

3.3.1 Case: The New Edge is Included in the Current Spanning Tree

Suppose that the edge $(\mathtt{u}, \mathtt{v})$ is an edge in $\mathtt{T}$. Then, since every vertex in $\mathtt{V}$ is included in $\mathtt{T}$ and every edge in $\mathtt{E_b}$ is included in $\mathtt{T}$ (because $\mathtt{G_b}$ is a subgraph of $\mathtt{T}$), it follows by the definition of $\mathtt{E'_b}$ (given at line (2), above) that $\mathtt{G'_b}$ is a subgraph of $\mathtt{T}$ in this case.

It therefore suffices to set $\mathtt{T'} = \mathtt{T}$ in this case in order to complete the proof.

3.3.2 Case: The New Edge is Not Included in the Current Spanning Tree

Suppose, instead, that the edge $(\mathtt{u}, \mathtt{v})$ is *not* an edge in $\mathtt{T}$. Recall again that, since $\mathtt{T}$ is a spanning tree of $\mathtt{G}$, $\mathtt{T}$ includes all the vertices that $\mathtt{G}$ does; let

$$\mathtt{T} = (\mathtt{V}, \mathtt{E_T}) \tag{3}$$

and consider the graph

$$\mathtt{U} = (\mathtt{V}, \mathtt{E_U}) \qquad \text{where } \mathtt{E_U} = \mathtt{E_T} \cup \{(\mathtt{u}, \mathtt{v})\}. \tag{4}$$

Claim 1. *$\mathtt{U}$ is a subgraph of $\mathtt{G}$ that contains a cycle.*

Proof. First, note that $\mathtt{U}$ includes the same set of vertices as $\mathtt{G}$. Since $\mathtt{E_T} \subseteq \mathtt{E}$ (because $\mathtt{T}$ is a subgraph of $\mathtt{G}$) and since $(\mathtt{u}, \mathtt{v}) \in \mathtt{E}$, $\mathtt{E_U} \subseteq \mathtt{E}$, so that $\mathtt{U}$ is a subgraph of $\mathtt{G}$.

It follows by Corollary 5 in the lecture notes on Trees, Spanning Trees, and Subgraphs that, since $\mathtt{T}$ is a spanning tree of $\mathtt{G}$, $|\mathtt{E_T}| = |\mathtt{V}| - 1$.

Now, since $(\mathtt{u}, \mathtt{v}) \notin \mathtt{E_T}$ it follows by the definition of $\mathtt{U}$, given at line (4) above, that $\mathtt{U}$ includes $|\mathtt{V}|$ vertices and $(|\mathtt{V}| - 1) + 1 = |\mathtt{V}|$ edges. It follows by Lemma 4 in the above set of notes that $\mathtt{U}$ contains a cycle, as claimed. □

Claim 2. *$\mathtt{U}$ contains a cycle that includes the edge $(\mathtt{u}, \mathtt{v})$.*

Proof. It follows by Claim 1, above, that $\mathtt{U}$ includes some cycle

$$(\mathtt{u}_1, \mathtt{u}_2), (\mathtt{u}_2, \mathtt{u}_3), \dots, (\mathtt{u}_{k-1}, \mathtt{u}_k), (\mathtt{u}_k, \mathtt{u}_1).$$

Suppose that the edge $(\mathtt{u}, \mathtt{v})$ is *not* one of these edges; then, since

$$\mathtt{E_U} = \mathtt{E_T} \cup \{(\mathtt{u}, \mathtt{v})\}$$

it follows that each of the edges $(\mathtt{u}_i, \mathtt{u}_{i+1})$ (for $1 \le i \le k-1$) and $(\mathtt{u}_k, \mathtt{u}_1)$ belongs to the set $\mathtt{E_T}$. Thus the above cycle is also a cycle in $\mathtt{T}$ — which is impossible, since $\mathtt{T}$ is a *tree.*

Thus the edge $(\mathtt{u}, \mathtt{v})$ is included the above cycle, so that $\mathtt{U}$ has a cycle including the edge $(\mathtt{u}, \mathtt{v})$, as claimed. □

Notice that, when listing the sequence of vertices in any cycle, we may begin with any one of the vertices that are be listed. Reversing the *order* of the listing of vertices (which is also possible for an undirected graph), if necessary, we may now assume, by an application of Claim 2, that $\mathtt{U}$ includes a cycle

$$(\mathtt{u}_1, \mathtt{u}_2), (\mathtt{u}_2, \mathtt{u}_3), \ldots, (\mathtt{u}_{k-1}, \mathtt{u}_k), (\mathtt{u}_k, \mathtt{u}_1) \tag{5}$$

such that $\mathtt{v} = \mathtt{u}_1$ *and* $\mathtt{u} = \mathtt{u}_2$. Recall that, since this is a cycle, $k \geq 3$. We may also assume (by considering the *shortest* cycle that begins with $\mathtt{v}$ and $\mathtt{u}$) that the vertices $\mathtt{u}_1, \mathtt{u}_2, \ldots, \mathtt{u}_k$ are *distinct* — that is, this is a "simple cycle."

Consider the colours of the vertices in this cycle at the *beginning* of the $k+1^{\text{st}}$ execution of the body of the loop: $\mathtt{u}_1 = \mathtt{v}$ is `grey` at this point because $\mathtt{v}$ is stored on the priority queue and, as discussed in the analysis of the algorithm, $\mathtt{u}_2 = \mathtt{u} = \pi(\mathtt{v})$ is a `black` node at this point.

Recall as well that the neighbours of all `black` nodes are either `black` or `grey`.

Considering the colours of each of the vertices $\mathtt{u}_3, \mathtt{u}_4, \ldots$ in turn, until a `grey` vertex is found, we see that exactly one of the following two cases must apply. Vertices $\mathtt{w}$ and $\mathtt{x}$ will be defined in each case.

1. The vertex $\mathtt{u}_i$ is `black` for $2 \leq i \leq k$, so that $\mathtt{v} = \mathtt{u}_1$ is the only `grey` node in this cycle. Let $\mathtt{w} = \mathtt{u}_k$ and let $\mathtt{x} = \mathtt{u}_1 = \mathtt{v}$ in this case.

2. There exists an integer ℓ such that $3 \leq \ell \leq k$, $\mathtt{u}_j$ is `black` for $2 \leq j \leq \ell - 1$, and such that $\mathtt{u}_\ell$ is `grey`. Let $\mathtt{w} = \mathtt{u}_{\ell-1}$ and let $\mathtt{x} = \mathtt{u}_\ell$ in this case.

Claim 3. *The vertices $\mathtt{w}$ and $\mathtt{x}$ satisfy each of the following properties.*

(a) The vertex $\mathtt{w}$ is `black` at the beginning of the $k+1^{st}$ execution of the body of the `while` loop.

(b) The vertex $\mathtt{x}$ is `grey` at the beginning of the $k+1^{st}$ execution of the body of the `while` loop.

(c) The edge $(\mathtt{w}, \mathtt{x})$ is not an edge in $\mathtt{G}'_{\mathtt{b}}$. That is, $(\mathtt{w}, \mathtt{x}) \notin \mathtt{E}'_{\mathtt{b}}$.

Proof. Properties (a) and (b) are easily established by a consideration of the definition of $\mathtt{w}$ and $\mathtt{x}$ in each of the two cases that are listed above.

Suppose that the first case is applicable, so that $\mathtt{x} = \mathtt{u}_1 = \mathtt{v}$ and $\mathtt{w} = \mathtt{u}_k$. Since $\mathtt{u}_1, \mathtt{u}_2, \ldots, \mathtt{u}_k$ are distinct, $k \geq 3$, $\mathtt{u} = \mathtt{u}_2$ and $\mathtt{w} = \mathtt{u}_k$, $\mathtt{w} \neq \mathtt{u}$.

Now notice that, since $\mathtt{v}$ is being added to the set of vertices to be included in the tree $\mathtt{G}'_{\mathtt{b}}$ during this execution of the loop body (at the same time as the edge $(\mathtt{u}, \mathtt{v})$ is being added), the degree of $\mathtt{v}$ in $\mathtt{G}'_\mathtt{b}$ is *one* and $\mathtt{u}$ is the *only* neighbour of $\mathtt{v}$ in $\mathtt{G}'_\mathtt{b}$. Since $\mathtt{w} \neq \mathtt{u}$ $\mathtt{w}$ cannot be a neighbour of $\mathtt{v}$ in $\mathtt{G}'_\mathtt{b}$ as well, so the edge $(\mathtt{w}, \mathtt{v})$ (which is the same as the edge $(\mathtt{w}, \mathtt{x})$) *does not* belong to $\mathtt{G}'_\mathtt{b}$. That is, property (c) holds.

Suppose, instead, that the second case is applicable. In this case $\mathtt{x} = \mathtt{u}_\ell$ is a **grey** node that is different from $\mathtt{v}$. Since $\mathtt{x}$ is grey at this point $\mathtt{x} \notin \mathtt{V}_\mathtt{b}$, and $\mathtt{x} \neq \mathtt{v}$, so $\mathtt{x} \notin \mathtt{V}'_\mathtt{b}$. It is clearly impossible for $(\mathtt{w}, \mathtt{x})$ to belong to $\mathtt{G}'_\mathtt{b}$ in this case. Thus property (c) also holds in this case as well. $\square$

We are now ready to define the minimum-cost spanning tree $\mathtt{T}'$ that will be used to complete the proof for this case: Let

$$\mathtt{T}' = (\mathtt{V}, \mathtt{E}'_\mathtt{T}) \qquad \text{where } \mathtt{E}'_\mathtt{T} = (\mathtt{E}_\mathtt{T} \setminus \{(\mathtt{w}, \mathtt{x})\}) \cup \{(\mathtt{u}, \mathtt{v})\} \tag{6}$$

That is, the edges in $\mathtt{T}'$ are all of the edges in $\mathtt{T}$ *except for* $(\mathtt{w}, \mathtt{x})$, together with the new edge $(\mathtt{u}, \mathtt{v})$.

Claim 4. *T' is a subgraph of G with $|V|$ vertices and $|V| - 1$ edges.*

Proof. It is clear by inspection of the definitions of $\mathtt{T}'$ and $\mathtt{U}$ (at lines (6) and (4), respectively) that $\mathtt{T}'$ is a subgraph of $\mathtt{U}$. Since $\mathtt{U}$ is a subgraph of $\mathtt{G}$ it is easy to show that $\mathtt{T}'$ is a subgraph of $\mathtt{G}$ as well.

Since the set of vertices in $\mathtt{T}'$ is $\mathtt{V}$ it is clear that there $|\mathtt{V}|$ edges in $\mathtt{T}'$.

To see that the number of edges in $\mathtt{T}'$ is $|\mathtt{V}| - 1$, note first that, since $(\mathtt{u}, \mathtt{v})$ is an edge in $\mathtt{G}'_\mathtt{b}$ while $(\mathtt{w}, \mathtt{x})$ is not an edge in this graph, the edges $(\mathtt{u}, \mathtt{w})$ and $(\mathtt{w}, \mathtt{x})$ are clearly distinct. It follows that $(\mathtt{w}, \mathtt{x})$ is an edge in $\mathtt{T}$, since this is an edge in $\mathtt{U}$ (by the choice of $\mathtt{w}$ and $\mathtt{x}$) and since $(\mathtt{u}, \mathtt{v})$ is the only edge in $\mathtt{U}$ that is *not* also an edge in $\mathtt{T}$. That is, $(\mathtt{w}, \mathtt{x}) \in \mathtt{E}_\mathtt{T}$.

On the other hand, $(\mathtt{u}, \mathtt{v}) \notin \mathtt{E}_\mathtt{T}$ under the case that we are now considering.

It now follows by the definition of $\mathtt{T}'$ at line (6) that

$$|\mathtt{E}_\mathtt{T}[| = (|\mathtt{E}_\mathtt{T}| - 1) + 1 = |\mathtt{E}_\mathtt{T}|,$$

and $|\mathtt{E}_\mathtt{T}| = |\mathtt{V}| - 1$ since $\mathtt{T}$ is a spanning tree of $\mathtt{G}$. $\square$

Claim 5. *T' is a connected graph.*

Proof. It is necessary and sufficient to prove that there is a path from $\mathtt{y}$ to $\mathtt{z}$ *consisting of edges in* $\mathtt{T}'$ for each pair of distinct vertices $\mathtt{y}, \mathtt{z} \in \mathtt{V}$.

Note that, since $\mathtt{T}$ is a spanning tree of $\mathtt{G}$, $\mathtt{T}$ is connected and includes the same set of vertices as $\mathtt{T}'$, so there is a path

$$\mathtt{y} = \mathtt{w}_1, \mathtt{w}_2, \ldots, \mathtt{w}_h = \mathtt{z} \tag{7}$$

from $\mathtt{y}$ to $\mathtt{z}$ in $\mathtt{T}$. That is, $(\mathtt{w}_i, \mathtt{w}_{i+1}) \in \mathtt{E}_\mathtt{T}$ for $1 \le i \le h-1$. We may also assume that this is a "simple path," that is, $\mathtt{w}_1, \mathtt{w}_2, \dots, \mathtt{w}_h$ are distinct.

Now, either the edge $(\mathtt{w}, \mathtt{x})$ is included in the above path, or it is not. These cases will be considered separately.

Case: $(\mathtt{w}, \mathtt{x})$ *is included in the above path:* Recall that $(\mathtt{w}, \mathtt{x})$ is one of the edges included in the cycle

$$\mathtt{u}_1, \mathtt{u}_2, \dots, \mathtt{u}_k$$

in the graph $\mathtt{U}$ that is introduced at line (5), above, and that all the edges in this cycle are distinct. Since $(\mathtt{w}, \mathtt{x})$ is the only edge included in $\mathtt{U}$ that is not also included in $\mathtt{T}'$, *all of the other edges* in this cycle can be used to form a path between $\mathtt{w}$ and $\mathtt{x}$ in $\mathtt{T}'$.

A path from $\mathtt{y}$ to $\mathtt{z}$ in $\mathtt{T}'$ can now be formed from the path shown at line (7) by replacing each occurrence of the edge $(\mathtt{w}, \mathtt{x})$ with the path between $\mathtt{w}$ and $\mathtt{x}$ in $\mathtt{T}'$ (in the appropriate direction) that has just been described.

Case: $(\mathtt{w}, \mathtt{x})$ *is not included in the above path:* In this case, each edge in this path is an edge included in $\mathtt{T}'$, so that the path shown at line (7) is a path from $\mathtt{y}$ to $\mathtt{z}$ in $\mathtt{T}'$.

Since $\mathtt{y}$ and $\mathtt{z}$ were arbitrarily chosen from $\mathtt{V}$ it follows that $\mathtt{T}'$ is connected, as claimed. □

Claim 6. *T' is a tree.*

Proof. This follows directly from Claims 4, 5, and from Lemma 7 in the online lecture notes on Trees, Spanning Trees, and Subgraphs. □

Claim 7. *T' is a spanning tree of G.*

Proof. $\mathtt{T}'$ is a subgraph of $\mathtt{G}$, by Claim 4. This subgraph is a tree, by Claim 6, and it includes all the vertices in $\mathtt{G}$. It is therefore a spanning tree of $\mathtt{G}$. □

Claim 8. $w((w, x)) \ge w((u, v))$.

Proof. Recall that $\mathtt{w}$ and $\mathtt{x}$ were defined in one of two ways, depending on the location of $\mathtt{w}$ in the cycle

$$\mathtt{u}_1, \mathtt{u}_2, \dots, \mathtt{u}_k$$

in $\mathtt{U}$ introduced at line (5) above. Each of these will be considered separately.

Case: $\mathtt{x} = \mathtt{u}_1 = \mathtt{v}$ *and* $\mathtt{w} = \mathtt{u}_k \ne \mathtt{u}_2 = \mathtt{u}$: Recall, from the analysis of this algorithm that $\mathtt{u} = \pi[\mathtt{v}]$,

$$\mathtt{w}((\mathtt{u}, \mathtt{v})) = \mathtt{d[v]} = \min_{\substack{\mathtt{y} \in \mathtt{V} \\ \mathtt{colour[y]=black} \\ \mathtt{(y,\ v) \in E}}} \mathtt{w}((\mathtt{y}, \mathtt{v})).$$

It follows by the choice of $\mathtt{w}$ (as described above) that the colour of $\mathtt{w}$ is `black` at the beginning of the $k+1^{\text{st}}$ execution of the body of the `while` loop and that $(\mathtt{w},\mathtt{v}) = (\mathtt{w},\mathtt{x}) \in \mathtt{E}$. It therefore follows by the above relationship that

$$\mathtt{w}((\mathtt{u},\mathtt{v})) = \mathtt{d[v]} = \mathtt{d[w]} \leq \mathtt{w}((\mathtt{w},\mathtt{x})),$$

as desired.

Case: $\mathtt{w} = \mathtt{u}_{\ell-1}$ *and* $\mathtt{x} = \mathtt{u}_\ell$ *where* $3 \leq \ell \leq k$. Recall that $\mathtt{w}$ is a `black` node and that $\mathtt{x}$ is a `grey` node that is different from $\mathtt{v}$, at the beginning of the $k+1^{\text{st}}$ execution of the body of the `while` loop, in this case.

Note that $\mathtt{w}((\mathtt{u},\mathtt{v})) = \mathtt{d[v]}$ in this case (and, for the same reason) once again. Since the element $\mathtt{v}$ is removed from the priority queue at the beginning of this execution of the body of the loop (with priority $\mathtt{d[v]}$ at this point), and since $\mathtt{x}$ is also stored on the priority queue with priority $\mathtt{d[x]}$, but *not* removed at this point, it is clear that $\mathtt{d[v]} \leq \mathtt{d[x]}$.

Finally, it follows by the notes on the analysis of this algorithm that, since $\mathtt{w}$ is `black` and $(\mathtt{w},\mathtt{x}) \in E$,

$$\mathtt{w}((\mathtt{w},\mathtt{x})) \geq \mathtt{d[x]} = \min_{\substack{\mathtt{y}\in\mathtt{V} \\ \mathtt{colour[y]=black} \\ (\mathtt{y},\,\mathtt{x})\in\mathtt{E}}} \mathtt{w}((\mathtt{y},\mathtt{x})).$$

Thus

$$\mathtt{w}(\mathtt{u},\mathtt{v}) = \mathtt{d[v]} \leq \mathtt{d[x]} \leq \mathtt{w}(\mathtt{w},\mathtt{x}))$$

in this case as well. $\square$

Claim 9. *T' is a minimum-cost spanning tree of $\mathtt{G}$.*

Proof. Since $\mathtt{T}'$ is a spanning tree of $\mathtt{G}$, by Claim 7, and since $\mathtt{T}$ is a minimum-cost spanning tree of $\mathtt{G}$, by assumption, $\mathtt{w}(\mathtt{T}) \leq \mathtt{w}(\mathtt{T}')$. It is necessary and sufficient to show that $\mathtt{w}(\mathtt{T}') \leq \mathtt{w}(\mathtt{T})$ as well — for then $\mathtt{w}(\mathtt{T}') = \mathtt{w}(\mathtt{T}')$ and *both* of these spanning trees are minimum-cost spanning trees of $\mathtt{G}$.

Recall that the set $\mathtt{E_T}$ of edges in $\mathtt{T}$ and the set $\mathtt{E}'_\mathtt{T}$ of edges in $\mathtt{T}'$ are related as follows (see, in particular, line (6) above):

$$\mathtt{E}'_\mathtt{T} = (\mathtt{E_T} \setminus \{(\mathtt{w},\mathtt{x})\}) \cup \{(\mathtt{u},\mathtt{v})\}.$$

Furthermore, $(\mathtt{w},\mathtt{x}) \in \mathtt{E_T}$.

Therefore

$$\begin{aligned}
\mathtt{w}(\mathtt{T}') &= \sum_{(\mathtt{y},\mathtt{z})\in \mathtt{E}'_{\mathtt{T}}} \mathtt{w}((\mathtt{y},\mathtt{z})) \\
&= \sum_{(\mathtt{y},\mathtt{z})\in \mathtt{E}_{\mathtt{T}}} \mathtt{w}((\mathtt{y},\mathtt{z})) - \mathtt{w}((\mathtt{w},\mathtt{x})) + \mathtt{w}((\mathtt{u},\mathtt{v})) && \text{(since } \mathtt{E}'_{\mathtt{T}} = (\mathtt{E}_{\mathtt{T}} \setminus \{(\mathtt{w},\mathtt{x})\}) \cup \{(\mathtt{u},\mathtt{v})\}\text{)} \\
&= \mathtt{w}(\mathtt{T}) - (\mathtt{w}((\mathtt{w},\mathtt{x})) - \mathtt{w}(\mathtt{u},\mathtt{v}))) \\
&\leq \mathtt{w}(\mathtt{T}) && \text{(since } \mathtt{w}((\mathtt{w},\mathtt{x})) \geq \mathtt{w}((\mathtt{u},\mathtt{v})) \text{ by Claim 8, above).}
\end{aligned}$$

□

Claim 10. *$\mathtt{G}'_{\mathtt{b}}$ is a subgraph of $\mathtt{T}'$.*

Proof. Since the set of vertices in $\mathtt{G}'_{\mathtt{b}}$ is a subset of $\mathtt{V}$ and $\mathtt{T}'$ includes all the vertices in $\mathtt{V}$, it is necessary and sufficient to show that every edge in $\mathtt{G}'_{\mathtt{b}}$ is an edge in $\mathtt{T}'$.

The set of edges in $\mathtt{G}'$ is

$$\begin{aligned}
\mathtt{E}'_{\mathtt{b}} &= \mathtt{E}_{\mathtt{b}} \cup \{(\mathtt{u},\mathtt{v})\} && \text{(as shown at line (2))} \\
&\subseteq \mathtt{E}_{\mathtt{T}} \cup \{(\mathtt{u},\mathtt{v})\} && \text{(since } \mathtt{G}_{\mathtt{b}} \text{ is a subgraph of } \mathtt{T}\text{, so that } \mathtt{E}_{\mathtt{b}} \subseteq \mathtt{E}_{\mathtt{T}}\text{).}
\end{aligned}$$

Since $\mathtt{E}'_{\mathtt{T}} = (\mathtt{E}_{\mathtt{T}} \cup \{(\mathtt{u},\mathtt{v})\}) \setminus \{(\mathtt{w},\mathtt{x})\}$, as shown at line (6), it is sufficient to establish that $(\mathtt{w},\mathtt{x}) \notin \mathtt{E}'_{\mathtt{b}}$ in order to complete the proof. This has already been established (and is included as part of Claim 3, above). □

It follows by Claims 9 and 10 that $\mathtt{G}'_{\mathtt{b}}$ is a subgraph of minimum-cost spanning tree of $\mathtt{G}$, as required to complete the proof in this case.