Math 3134 Homework 5 Solutions

Mar 5, 2004

3.3.21. If $L(U) = \infty$, then $L(V) = \infty$ for all $V \notin P$. So there is no edge from any vertex in P to any vertex not in P. Hence there cannot be a path from S to any vertex not in P. In particular, there is no path from S to U. So saying that any shortest path from S to U contains only vertices in P before U is true by default.

If $L(U) < \infty$, then there exists a path p from S to U whose vertices before U are all in P and whose length is $L(U)$. Let q be a shortest path from S to U. Obviously $\text{length}(q) \leq \text{length}(p)$. Suppose q contains at least one vertex not in P before U. Among all such vertices, let X be the first one on the way from S to U. Let r be the part of q from S to X. Since the weights are positive, the part of q from X to S has positive length, so $\text{length}(r) < \text{length}(q)$. All the vertices of r other than X are in P, therefore $L(X) \leq \text{length}(r)$.

Putting these inequalities together:

$$L(X) \leq \text{length}(r) < \text{length}(q) \leq \text{length}(p) = L(U),$$

which contradicts the minimality of $L(U)$. Hence q cannot contain a vertex not in P other than U.

(Based on Steve Rugh's and Matthew Peterson's solutions.)

Note: Some of you claimed that property (ii) says that for any vertex $V \notin P$ a shortest path from S to V contains only vertices in P. This is false. Property (ii) says that among all paths from S to V whose vertices other than V are in P, $L(V)$ is the length of a shortest one.

This is a rather bad logical mistake and is comparable to the following. Your friend shows you her Peking dog, and says that of all the dogs she has owned, this is the largest. You conclude that the largest dog in existence is your friend's and is a Peking dog.

3.3.22. If $L(U) < \infty$, then there exists a shortest path q from S to U, and as we proved in the previous exercise, all of the vertices in q other than U are in P, therefore $\text{length}(q) = L(U)$.

If $L(U) = \infty$, then there is no path from S to U (see the proof of 3.3.21), so we may say that the length of a shortest path from S to U is ∞.

3.3.23. For a vertex V in G, let $D(V)$ its distance from S, that is the length of a shortest path from S to V, or ∞ is there is no path from S to V. (NB: I am not saying we know how to compute $D(V)$, only that it exists.)

We already know that for every vertex V in P, $L(V) = D(V)$ by condition (i) in 3.3.21. We also know $L(U) = D(U)$ from 3.3.22. So $P' = P \cup \{U\}$ satisfies property (i).

We will now prove property (iii) for $P' = P \cup \{U\}$. Let V be any vertex in P' and W any vertex not in P'. Then $W \notin P$ either. If $V \in P$ then $D(V) \leq D(W)$ by property (iii) for P. It remains to show that $D(U) \leq D(W)$. If $D(W) = \infty$, this is obviously true. If $D(W) < \infty$, then there is a shortest path q from S to W of length $D(W)$. Choose the first vertex X on q that is not in P (X could be W itself, or even U) and let r be the part of q from S to X. Just like in the proof of 3.3.21, $L(X) \leq \text{length}(r)$ and $\text{length}(r) \leq \text{length}(q) = D(W)$. (This time the inequality is not strict because we did not assume X is before W on q.) But U was chosen so $L(U)$ was minimal, hence

$$D(U) = L(U) \leq L(X) \leq \text{length}(r) \leq D(W).$$

So P' satisfies (iii).

Observe that property (iii) for P implies that for $V \in P$, a shortest path from S to V contains only vertices in P. If it contained any vertex $X \notin P$, then the part of the path from S to X would be of length strictly less than $D(V)$, but then $D(X) < D(V)$.

Let $V \notin P'$. Now we will show that the length of a shortest path from S to V whose vertices other than V are all in P' is the minimum of $L(V)$ and $L(U) + W(U, V)$.

If there is no such path then there is also no path from S to V whose vertices other than V are all in P, hence $L(V) = \infty$. In this case, there is also no edge from U to V, hence $L(U) + W(U, V) = L(U) + \infty = \infty$. Then we would say the length of a shortest path from S to V whose vertices other than V are all in P' is ∞, which is indeed the minimum of $L(V)$ and $L(U) + W(U, V)$.

Otherwise, let p be a shortest path from S to V whose vertices other than V are all in P'. If U is not on p, then all the vertices of p other than V are in P, so length$(p) = L(V)$. Otherwise U is on p. If so, U must be the last vertex in P' before V. If it were not, then the vertex W which follows U on p is in P. Let q be the part of p from S to W. By the observation above, q cannot be a shortest path from S to W because it contains $U \notin P$. So there is a shorter path from S to W all of whose vertices are in $P \subset P'$, which would contradict the choice of p as a shortest path from S to V all of whose vertices other than V are in P'. Hence U must be the last vertex before V, and hence the length$(p) = L(U) + W(U, V)$.

If you want a shortest path from S to V whose vertices other than V are in P', you pick the shorter of the above two choices.

Note: Many of you tried to prove property (iii) for P' by pointing out that $L(U) \leq L(W)$ for all $W \notin P$. This is not enough because for $W \notin P$, $L(W)$ is not the distance of W from S.

3.3.24. The hardest part in this exercise seemed to be to figure out what Dijkstra's algorithm had to do with the previous three problems. Dijkstra's algorithm is an inductive process: it keeps adding vertices to the set P one by one, and it keeps updating the labels on the vertices outside P if it finds a shorter path to those from S than previously known. 3.3.21 and 22 reflect how Dijkstra's algorithm chooses the next vertex to add to P. 3.3.23 shows that adding U to P maintains properties (i) and (iii) and updating the labels of the vertices not in P with $\min\{L(V), L(U) + W(U, V)\}$ restores property (ii). So we can repeat choosing U, adding it to P and updating the labels until we run out of vertices, and it will always be true for the vertices in P that their labels are their distances from S. Here is the formal proof:

The base case of the induction is the initialization in Dijkstra's algorithm. It sets up $P = \{S\}$, $L(S) = 0$ and $L(v) = W(S, V)$ for all $V \neq S$. This clearly satisfies (i) and (iii). Notice that for $V \notin P$ the only possible path from S to V whose vertices other than V are in P is an edge from S to V, if there is one. So this initial setup satisfies (ii) as well.

Suppose that the set P and the labels L in the k-th step in Dijkstra's algorithm (that is when P has k elements) satisfy properties (i), (ii), and (iii).

Observe that the algorithm says to choose U exactly as in 3.3.21 and to replace P with $P' = P \cup \{U\}$. As shown in 3.3.23, P' satisfies (i) and (iii). Then it says to update the labels on the vertices not in P' with $L(U) + W(U, V)$ if this is less than the old label, $L(V)$. We proved in 3.3.23 that after this update, P' satisfies (ii). So in the $k + 1$-st step in Dijkstra's algorithm, P and L still satisfy (i), (ii), and (iii).

Dijkstra's algorithm terminates when all of the vertices have been added to P. By property (i), the label on any vertex V at this stage is the length of a shortest path from S to V.

(Based on John Bowen's solution.)