Australian National University
Department of Computer Science
Week 12

Computer Science COMP3600/COMP6466 in 2009 – Answer to Tutorial 4

Question 1. *A depth-first forest classifies th edges of a graph into tree, back, forward, and cross edges. A breadth-first tree can also be used to classify the edges reachable from the source of the search into the same four categories. Prove that in a breadth-first search of a directed graph, the following properties holds:*

1. *There are no forward edges.*
2. *For each tree edge $\langle u, v \rangle$, we have $d[v] = d[u] + 1$.*
3. *For each cross edge $\langle u, v \rangle$, we have $d[v] \le d[u] + 1$.*
4. *For each back edge $\langle u, v \rangle$, we have $0 \le d[v] \le d[u]$.*

Answer: (1) Assume that there is a forward edge $\langle u, v \rangle$ and u is an ancestor of v in the BFS tree. Then, when u is in the queue (a queue contains gray nodes only), the color of v has one of the three possibilities: white, gray, and black.

If v is white, then, the edge $\langle u, v \rangle$ is a tree edge when u is the head of the queue and u explores all its neighboring nodes, this contradicts that this edge is a forward edge.

If v is black, then, v has been visited before, and u has not yet been colored black, which means there is no relationship between u and v. This contradicts the fact that $\langle u, v \rangle$ is a forward edge and nodes u and v are the ancestor-descendent relationship.

Thus, v must be gray already. Now u explores its neighboring nodes including v, which means that there is no relationship between u and v. This contradicts the fact that the edge $\langle u, v \rangle$ is a forward edge.

In summary, there is no forward edge in BFS search in a directed graph.

(2) Following the BFS tree construction, if $\langle u, v \rangle$ is a tree edge, then $d[v] = d[u] + 1$, because u is the parent of v in the tree.

(3) Consider a cross edge $\langle u, v \rangle$. Assume that $d[v] > d[u] + 1$, i.e., $d[v] \ge d[u] + 2$. When u becomes the head of the queue consisting of gray nodes, its neighboring nodes will be explored, then, $d[v] \le d[u] + 1$, while we already knew that $d[v] \ge d[u] + 2$, which contradicts that the tree is a BFS tree. Thus, $d[v] \le d[u] + 1$.

(4) Assume that $\langle u, v \rangle$ is a back edge, i.e, v is an ancestor of u in the BFS tree, then $d[v] \le d[u]$ following the definition of the BFS tree. it is obvious that $d[v] > 0$.

Question 2. *(a) There are two well-known algorithms for finding minimum spanning trees. Point out the difference between Prim's algorithm and Kruskal's algorithm in terms of the construction of the MST tree.*

(b) Assume that the weight assigned to each edge in graph $G = (V, E)$ is distinct. Let e be a maximum-weighted edge on a cycle of $G = (V, E)$. Show that a minimum spanning tree in $G' = (V, E - \{e\})$ is also a minimum spanning tree in G.

Answer: (a) The construction of an MST tree by Prim's algorithm is started from a single node, and the tree is expanded until it covers all the nodes in the graph. In other words, there is only one partial tree during the construction of the MST tree from the very beginning to the end. However, in the construction of an MST tree using Kruskal's algorithm, the algorithm starts from a n-tree forest. Each time it merges two trees in the forest into one, the algorithm continues until there is only one tree left in the forest.

(b) Let $T(V, E(T))$ be an MST of $G(V, E)$, then $e \notin E(T)$, because otherwise it is easy to show that T is not an MST of G.

The only difference between $G(V, E)$ and $G(V, E - \{e\})$ is the edge e. We claim that $e \notin E(T')$ as well, where $T'(V, E(T'))$ is an MST of $G(V, E - \{e\})$. Assume that this is not true. Obviously, T is a spanning tree of $G(V, E - \{e\})$, $w(T) \geq w(T')$, this is a contradiction that T is an MST of G.

Question 3. *Given a directed weighted graph $G(V, E)$ with source s, in which case the Bellman-Ford Algorithm instead of Dijkstra's algorithm should be used to solve the single-source shortest paths problem with source s.*

Answer: When the weight associated with an edge is negative, the Bellman-Ford algorithm instead of Dijkstra's algorithm should be used for finding the single source shortest paths problem with source at s, simply because that Dijkstra's algorithm is inapplicable in this case.

Question 4. *Given a directed weighted graph $G(V, E)$, let $d(v_i, v_j)$ be the distance in G from v_i to v_j in terms of the minimum number of edges used in the path, $v_i \in V$ and $v_j \in V$. Define the diameter D of G as $D = \max_{v_i \in V, v_j \in V}\{d(v_i, v_j)\}$. What is the time complexity for finding all pairs of shortest paths in G if the repeated squaring approach is applied?*

Answer: Let M be a weighted adjacency matrix of G. In order to find all pairs of shortest paths in G, we usually have to calculate

$$M^{(1)}, M^{(2)}, M^{(3)}, \ldots, M^{(n-1)},$$

where $M^{(i+1)} = M^{(i)} * M$.

If we use the squaring approach, we only need to calculate

$$M^{(1)}, M^{(2)}, M^{(4)}, \ldots, M^{2^{\lceil \log(n-1) \rceil}},$$

where $M^{(2i)} = M^{(i)} * M^{(i)}$ and $(n-1)$ is the maximum number of edges in a shortest path.

If D is given, we only need to calculate

$$M^{(1)}, M^{(2)}, M^{(4)}, \ldots, M^{2^{\lceil \log D \rceil}}.$$

The running time of the repeated squaring algorithm thus is $O(n^3 \log D)$, where the $O(n^3)$ factor is the time of two matrix multiplication of order n.