

25-1 *Transitive closure of a dynamic graph*

Suppose that we wish to maintain the transitive closure of a directed graph $G = (V, E)$ as we insert edges into E . That is, after each edge has been inserted, we want to update the transitive closure of the edges inserted so far. Assume that the graph G has no edges initially and that we represent the transitive closure as a boolean matrix.

- a. Show how to update the transitive closure $G^* = (V, E^*)$ of a graph $G = (V, E)$ in $O(V^2)$ time when a new edge is added to G .
- b. Give an example of a graph G and an edge e such that $\Omega(V^2)$ time is required to update the transitive closure after the insertion of e into G , no matter what algorithm is used.
- c. Describe an efficient algorithm for updating the transitive closure as edges are inserted into the graph. For any sequence of n insertions, your algorithm should run in total time $\sum_{i=1}^n t_i = O(V^3)$, where t_i is the time to update the transitive closure upon inserting the i th edge. Prove that your algorithm attains this time bound.

26.1-4

Let f be a flow in a network, and let α be a real number. The *scalar flow product*, denoted αf , is a function from $V \times V$ to \mathbb{R} defined by

$$(\alpha f)(u, v) = \alpha \cdot f(u, v) .$$

Prove that the flows in a network form a *convex set*. That is, show that if f_1 and f_2 are flows, then so is $\alpha f_1 + (1 - \alpha) f_2$ for all α in the range $0 \leq \alpha \leq 1$.

26.2-4

In the example of Figure 26.6, what is the minimum cut corresponding to the maximum flow shown? Of the augmenting paths appearing in the example, which one cancels flow?

34.2-2

Prove that if G is an undirected bipartite graph with an odd number of vertices, then G is nonhamiltonian.

34.2-6

A *hamiltonian path* in a graph is a simple path that visits every vertex exactly once. Show that the language $\text{HAM-PATH} = \{\langle G, u, v \rangle : \text{there is a hamiltonian path from } u \text{ to } v \text{ in graph } G\}$ belongs to NP.

34.5-7

The *longest-simple-cycle problem* is the problem of determining a simple cycle (no repeated vertices) of maximum length in a graph. Formulate a related decision problem, and show that the decision problem is NP-complete.