Fundamental Algorithms

Homework 10

Due Wednesday, Nov. 28 at the beginning of class

Read chapter 8 through 8.3.3.

8.0 a) Without looking at the book, write a version of the Floyd-Warshall algorithm that includes path recovery.

b) Without looking at the book, present an algorithm PathPrint(i, j) that uses the solution to part a to print the consecutive sequence of vertices on a shortest path from i to j.

8.05 Given a directed graph G and edgeweight function $EdgCst(i, j)$ present an algorithm that, for each pair of vertices (i, j), computes the length of the shortest path from i to j that has exactly two edges. You can assume that $EdgCst(i, j) = \infty$ if G does not have an edge from i to j.

* 8.1 Write an enhancement to the Floyd-Warshall algorithm that saves, in Intermediate[i; j], the first vertex after i, on the shortest path from i to j. Notice that proper initialization makes the algorithm simpler. Hint: There is a fundamental principle of iterative, inductive, and recursive algorithm design that makes the solution easier. What is this principle of life?

Hint: Believe in your partial solution, and use that partial solution as a guiding principle to focus your thoughts, and to determine the next step in the algorithm.

* 8.2 Write an enhancement to the Floyd-Warshall algorithm that saves, in Intermediate[i; j], the last vertex before j, on the shortest path from i to j. Notice that proper initialization makes the algorithm simpler.

8.3 Some students mistakenly transpose, in the Floyd-Warshall algorithm, the lines

15 for k ← 1 to n do
16 for i ← 1 to n do
17 for j ← 1 to n do
: : :

to read,

15 for i ← 1 to n do
16 for j ← 1 to n do
17 for k ← 1 to n do
: : :

Let the transposed algorithm be run on a general graph. What paths will it consider as possible candidates for the shortest path from vertex 1 to vertex 2? Describe this selection in words.

8.21 a) Show how to modify the Floyd-Warshall algorithm to compute the shortest path lengths in a graph where the length of each path is defined to equal the length of the longest edge on the path (instead of the sum of edge lengths).

b) Show how to modify the Floyd-Warshall algorithm to compute the shortest path lengths in a graph where the length of each path is defined to equal the product of the lengths of the edges on the path (instead of the sum of edge lengths). A technical requirement for the original problem is that the graph have no negative cycles. For this product version, you can assume that all edges have positive lengths, but there is still a technical requirement about cycles. What is that requirement, and why?

** 8.41 Modify the Floyd-Warshall algorithm to solve the All-Pairs-First-and-Second-Shortest-Paths problem, which computes the lengths of the shortest and second shortest paths between each pair of vertices. Note that two paths are different if they are not the exact same sequence of edges. For specificity, let the value for second shortest path between a pair of vertices be the same as the shortest if there are two paths between the vertices that both have the shortest possible length.

Hint: You need to draw an abstract picture, and to think clearly to cover all of the cases. You also need to time your updates carefully.

* 8.42 Modify the Floyd-Warshall algorithm to solve the All-Pairs-Shortest-Paths problem and to compute the number of shortest paths between each pair of vertices, so that if there are seven equally short (shortest) paths between a pair of vertices, you algorithm should detect this fact.

Hints

Suppose there are two shortest paths from i to k, and three shortest paths from k to j. How many shortest paths are there from i to j that go through k?

Hint: Did you say the product of two and three?

Suppose that there were seven other shortest paths, so that they all have the same length. How many shortest paths are there altogether?

Hint: Did you say thirteen? Did you increment any counters to compute these answers? (No)