CSE 326: Data Structures
Dynamic Programming –
Floyd/Warshall Algorithm

Hal Perkins
Spring 2007
Lectures 26

Analysis

- Total running time for Dijkstra’s:
  \( O(|V|^2 + |E|) \) (linear scan)
  \( O(|V| \log |V| + |E| \log |V|) \) (heaps)

  What if we want to find the shortest path from each point to ALL other points?

Single-Source Shortest Path

- Given a graph \( G = (V, E) \) and a single distinguished vertex \( s \), find the shortest weighted path from \( s \) to every other vertex in \( G \).

All-Pairs Shortest Path:

- Find the shortest paths between all pairs of vertices in the graph.
- How?

Dynamic Programming

Algorithmic technique that systematically records the answers to sub-problems in a table and re-uses those recorded results (rather than re-computing them).

Simple Example: Calculating the Nth Fibonacci number.

\( \text{Fib}(N) = \text{Fib}(N-1) + \text{Fib}(N-2) \)
Floyd-Warshall

for (int k = 1; k <= V; k++)
for (int i = 1; i <= V; i++)
for (int j = 1; j <= V; j++)
if ( ( M[i][k]+ M[k][j] ) < M[i][j] )
   M[i][j] = M[i][k]+ M[k][j]

Invariant: After the kth iteration, the matrix includes the shortest paths for all pairs of vertices (i,j) containing only vertices 1..k as intermediate vertices

Initial state of the matrix:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>2</td>
<td>-</td>
<td>-4</td>
<td>-</td>
</tr>
<tr>
<td>b</td>
<td>-</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>c</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>e</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

M[i][j] = min(M[i][j], M[i][k]+ M[k][j])

Floyd-Warshall - for All-pairs shortest path

Final Matrix Contents