Problem Set 7

Reading Assignment

Read:

• Read Chapter 11, *Graphs*.


Goals

Practice programming with graphs. Implementation of a complicated algorithm, top-down design.

Assignment

Implement Dijkstra’s shortest-path algorithm. The program accepts the commands,

• **Go** — prompts for the source vertex then runs the shortest-path algorithm.

• **List** — prints out the current graph.

• **Answer** — Prints out the answer to the shortest-path problem. For each vertex $n$ give the length of the shortest-path from the source to $n$ and the parent of $n$ along the shortest path.

• **Quit** — exits the program.

• $n \ m \ w$, where $n$, $m$ and $w$ are positive integers. This means: make an edge of weight $w$ between vertices $n$ and $m$. 
Example

>2 3 1
>3 1 1
>1 2 3
> List

Graph:

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Vertex</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

> Go
=> Enter source: 2
Computing Shortest-paths from 2 ...
Done.
> Answer

Shortest-Paths

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Parent</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

> Quit

Extra Credit

For extra credit, implement the command How?, which prompts for a vertex n then gives the entire shortest path from the source to n. After the Answer command of the previous example, the How? command would give this output:

> How?
=> Enter vertex: 1
  2 -> 3 -> 1 : length=2