Dijkstra’s Algorithm

Examples

Dijkstra’s Algorithm: Pseudocode

- Initialize the cost of each node to \( \infty \)
- Initialize the cost of the source to 0
- While there are unknown nodes left in the graph
  - Select an unknown node \( b \) with the lowest cost
  - Mark \( b \) as known
  - For each node \( a \) adjacent to \( b \)
    - If \( b \)'s cost + cost of (\( b, a \)) < \( a \)'s old cost
      - \( a \)'s cost = \( b \)'s cost + cost of (\( b, a \))
      - \( a \)'s prev path node = \( b \)

Important Features

- Once a vertex is made known, the cost of the shortest path to that node is known
- While a vertex is still not known, another shorter path to it might still be found
- The shortest path itself can be found by following the backward pointers stored in node.path

Notes on these examples

- These examples use:
  - ?? instead of infinity.
  - Visited? instead of Known
  - Found By instead of Path

Dijkstra’s Example

<table>
<thead>
<tr>
<th>V</th>
<th>Visited?</th>
<th>Cost</th>
<th>Found by</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0</td>
<td>Y</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>v1</td>
<td>??</td>
<td>2</td>
<td>v0</td>
</tr>
<tr>
<td>v2</td>
<td>&lt;= 2</td>
<td>v0</td>
<td></td>
</tr>
<tr>
<td>v3</td>
<td>&lt;= 1</td>
<td>v0</td>
<td></td>
</tr>
<tr>
<td>v4</td>
<td>??</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>v5</td>
<td>??</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>v6</td>
<td>??</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>
At this point either v2 or v4 would be fine to pick next.
Dijkstra’s Algorithm in action

At this point either D or F would be fine to pick next.
Dijkstra’s Algorithm in action

Vertex Visited? Cost Found by
A Y 0 A
B Y 2 A
C Y 1 A
D Y 4 A
E <=11 C
F Y 4 B
G <=8 H
H Y 7 F

Final Answer

Vertex Visited? Cost Found by
A Y 0 A
B Y 2 A
C Y 1 A
D Y 4 A
E Y 11 G
F Y 4 B
G Y 6 H
H Y 7 F