Dijkstra’s Algorithm (Single Source Shortest Path)

- The overall logic is the same as Prim’s Algorithm
- We will modify the code in only two places – both involving the update to the distance metric.
- The result is a directed acyclic graph or DAG

Backtracking in Dijkstra

Dijkstra’s Algorithm gives us the shortest path from a single source to every connected vertex:

The data structure maintained by Dijkstra’s Algorithm will have the following state after running Dijkstra’s Algorithm:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>NULL</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>F</td>
<td>A</td>
<td>F</td>
<td>C</td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>10</td>
<td>17</td>
<td>15</td>
<td>12</td>
<td>7</td>
<td>11</td>
<td>21</td>
</tr>
</tbody>
</table>

Q: What is the shortest path from A to H?

Q: What is the shortest path from A to E?
**Examples:** How is a single heavy-weight path vs. many light-weight paths handled?

Ex 1:

![Diagram of a graph with weights](image)

Ex 2:

![Diagram of a graph with weights](image)

What about undirected graphs?

![Diagram of an undirected graph with weights](image)

**Dijkstra:** What if we have a minimum-weight edge, without having a negative-weight cycle?

![Diagram of a graph with weights](image)

Dijkstra makes an assumption:

**Dijkstra Algorithm:** What is the running time?

**Challenge:** Landmark Path Problem

![Diagram of a graph with landmarks and weights](image)

**CS 225 – Things To Be Doing:**

1. Final Exam runs Thursday, May 3 – Thursday, May 10
2. MP7 is released; MP7 deadline Monday, May 30
3. Final lab, `lab_ml`, released today; due Sunday, May 29
4. This week is the last week of POTDs (last POTD is Friday!)