Dijkstra’s Algorithm Overview:

- 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 containing the shortest path to every vertex from a single starting point.

### Pseudocode for Dijkstra’s SSSP Algorithm

```plaintext
DijkstraSSSP(G, s):
    foreach (Vertex v : G):
        d[v] = +inf
        p[v] = NULL
    d[s] = 0
    PriorityQueue Q // min distance, defined by d[v]
    Q.buildHeap(G.vertices())
    Graph T // "labeled set"
    repeat n times:
        Vertex m = Q.removeMin()
        T.add(m)
        foreach (Vertex v : neighbors of m not in T):
            if __________________________ < d[v]:
                d[v] = ________________
                p[v] = m
    return T
```

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

CS 225 Final Exam
- The final exam begins on Thursday, May 3\textsuperscript{rd}
- The final exam is a 3 hour CBTF exam, is a cumulative exam, and has the format of a combined theory + programming exam
- The last office hours is Wednesday, May 2\textsuperscript{nd}
- \textit{We'll use lecture on Wednesday, May 2\textsuperscript{nd} as a final exam review!}

“Pre-Final” Grade Dump
- I believe there’s only a few remaining issues left with grading;
  - I’ll be starting to wrap these up myself over the weekend:
    - +EC from creative components
    - Working on recovering repos that were force deleted
- As soon as possible after MP7’s deadline, we’ll provide a “Pre-Final” grade in Compass that incorporates everything except the final exam into your CS 225 grade.

End of Semester Grade Review
- Excel sheet will be provided once final grades are posted.
- Must submit an Excel sheet for this review.

Floyd-Warshall Algorithm
Floyd-Warshall’s Algorithm is an alternative to Dijkstra in the presence of negative-weight edges (but \textbf{not} negative weight cycles).

\begin{verbatim}
Pseudocode for Floyd-Warshall’s Algorithm
1 FloydWarshall(G):
2 Input: G, Graph;
3 Output: d, an adjacency matrix of distances between all vertex pairs
4 let d be a adj. matrix initialized to +inf
5 foreach (Vertex v : G):
6 d[v][v] = 0
7 foreach (Edge (u, v) : G):
8 d[u][v] = cost(u, v)
9 foreach (Vertex u : G):
10 foreach (Vertex v : G):
11 foreach (Vertex w : G):
12 if d[u, v] > d[u, w] + d[w, v]:
13 d[u, v] = d[u, w] + d[w, v]
14 return d
\end{verbatim}

Running Floyd-Warshall:

\begin{verbatim}
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
\end{verbatim}

CS 225 – Things To Be Doing:

1. Final Exam runs Thursday, May 3 – Thursday, May 10
2. MP7 deadline Monday, April 30
3. Final lab, \texttt{lab_ml} due Sunday, April 29
4. Final POTD is right now! 😊