Dec. 8 – Dijkstra’s + Shortest Path
Wade Fagen-Ulmschneider
Dijkstra’s Algorithm (SSSP)

DijkstraSSSP(G, s):
6    foreach (Vertex v : G):
7        d[v] = +inf
8        p[v] = NULL
9        d[s] = 0
10       PriorityQueue Q // min distance, defined by d[v]
11       Q.buildHeap(G.vertices())
12       Graph T // "labeled set"
13       repeat n times:
14          Vertex u = Q.removeMin()
15          T.add(u)
16          foreach (Vertex v : neighbors of u not in T):
17              if cost(u, v) + d[u] < d[v]:
18                  d[v] = cost(u, v) + d[u]
19                  p[v] = m
20
21
Dijkstra’s Algorithm (SSSP)

Dijkstra gives us the shortest path from our path (single source) to every connected vertex!
Dijkstra’s Algorithm (SSSP)

Q: How does Dijkstra handle a single heavy-weight path vs. many light-weight paths?
Q: How does Dijkstra handle a single heavy-weight path vs. many light-weight paths?
Dijkstra’s Algorithm (SSSP)

Q: How does Dijkstra handle undirected graphs?
Dijkstra’s Algorithm (SSSP)

Q: How does Dijkstra handle negative weight cycles?
Q: How does Dijkstra handle negative weight edges, without a negative weight cycle?
Dijkstra’s Algorithm (SSSP)

What is Dijkstra’s running time?

DijkstraSSSP(G, s):
  6    foreach (Vertex v : G):
  7        d[v] = +inf
  8        p[v] = NULL
  9    d[s] = 0
  10
  11    PriorityQueue Q // min distance, defined by d[v]
  12    Q.buildHeap(G.vertices())
  13    Graph T      // "labeled set"
  14
  15    repeat n times:
  16       Vertex u = Q.removeMin()
  17       T.add(u)
  18       foreach (Vertex v : neighbors of u not in T):
  19          if cost(u, v) + d[u] < d[v]:
  20             d[v] = cost(u, v) + d[u]
  21             p[v] = u
  22
  23    return T
CS 225 – Office Hours

Open Lab Hours:
  Approx. ~40 hours of open lab hours left this semester!
  Last day of office hours: Wednesday, Dec. 13

Professor Office Hours (DYB):
  Mattox: Today from 12:15pm – 1:45pm, 3034 ECEB
  Wade: Next Wednesday from 12:15pm – 1:45pm, 3034 ECEB

Open Door Hours (no CS225 content):
  Wade: Today from 3:30pm – 5pm+, 2215 Siebel Center
CS 225 – Course Assistants for Spring 2018!

CS 225 will be expanding our CA team!
Help out in lab, office hours, piazza.
Help out on developing CS 225 resources
(including: infrastructure, guidebook, MPs, labs, etc.)

Reach out to us for an application:

E-mail Thierry if you’re interested!
Floyd-Warshall Algorithm

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

FloydWarshall(G):
6   Let d be a adj. matrix initialized to +inf
7   foreach (Vertex v : G):
8       d[v][v] = 0
9   foreach (Edge (u, v) : G):
10      d[u][v] = cost(u, v)
11
12   foreach (Vertex u : G):
13      foreach (Vertex v : G):
14         foreach (Vertex w : G):
15             if d[u, v] > d[u, w] + d[w, v]:
16                 d[u, v] = d[u, w] + d[w, v]
Floyd-Warshall Algorithm

FloydWarshall(G):
6 Let d be a adj. matrix initialized to +inf
7 foreach (Vertex v : G):
8 d[v][v] = 0
9 foreach (Edge (u, v) : G):
10 d[u][v] = cost(u, v)
11
12 foreach (Vertex u : G):
13 foreach (Vertex v : G):
14 foreach (Vertex w : G):
15 if d[u, v] > d[u, w] + d[w, v]:
16 d[u, v] = d[u, w] + d[w, v]

<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>
What is Floyd-Warshall’s running time?

```plaintext
FloydWarshall(G):
6   Let d be a adj. matrix initialized to +inf
7   foreach (Vertex v : G):
8       d[v][v] = 0
9   foreach (Edge (u, v) : G):
10      d[u][v] = cost(u, v)
11
12   foreach (Vertex u : G):
13       foreach (Vertex v : G):
14           foreach (Vertex w : G):
15               if d[u, v] > d[u, w] + d[w, v]:
16                   d[u, v] = d[u, w] + d[w, v]
```
MST Algorithm Runtime:

- Dijkstra’s Algorithm: \(O(m + n \log(n))\)

All Pairs Shortest Path:

- Dense Graphs: \(O(n^3)\)
- Sparse Graphs: \(O(n^3)\)
CS 225 – Things To Be Doing

Exam 13: Makeup Exam starts Monday
More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP7: The final MP!
Due: Monday, Dec. 11 at 11:59pm

Lab: The final lab, lab_ml, is due Sunday
lab_ml: Due Sunday @ 11:59pm

New POTDs every M/W/F
Worth +1 Extra Credit /problem (up to +40 total)