Programming, Problem Solving, and Algorithms

CPSC203, 2023 W2

Announcements

- POTW 10 due date extended to next week and POTW 11 due date extended to the week after.
    - No submissions can be accepted past April 11th (last day of the term) !

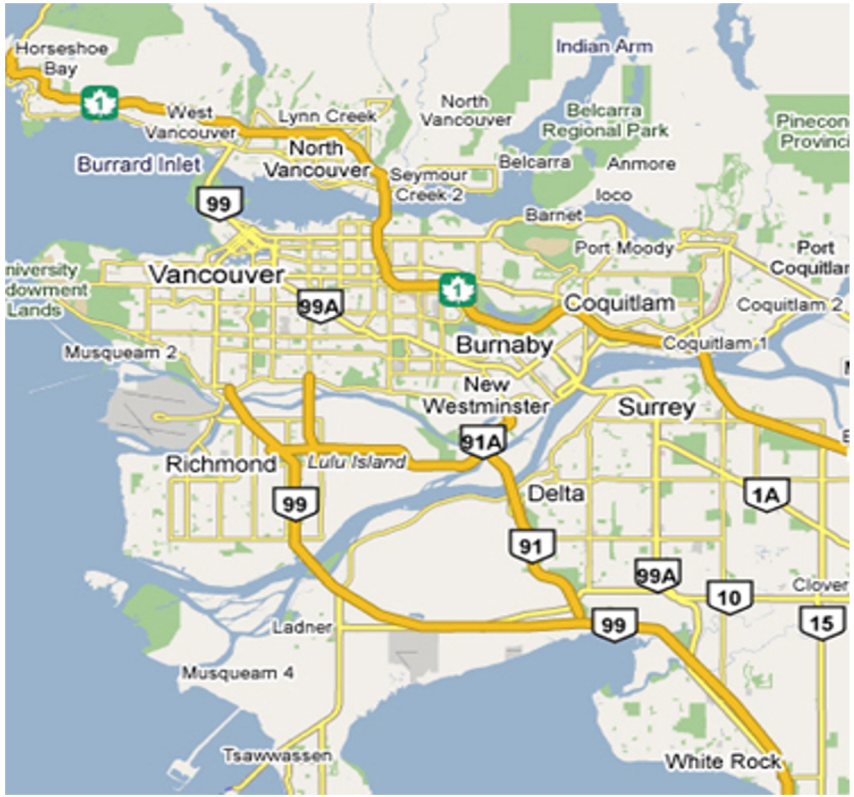
- Project 3 is on Maps!
    - Beta version: https://classroom.github.com/a/0jTxpPVD
    - We're still working on getting the autograder working for this, but the code should be stable.
    - You will have a little bit of coding to do, but the majority of it will be squashing bugs!
    - It will be due on the last day of the term.

Today's Plan...

1. Announcements! (10 mins)
2. Weekly Videos Review/Questions (20 mins)
3. Single Source Shortest Path (40 mins)

Slides from the Assigned Videos

Single Source Shortest Path

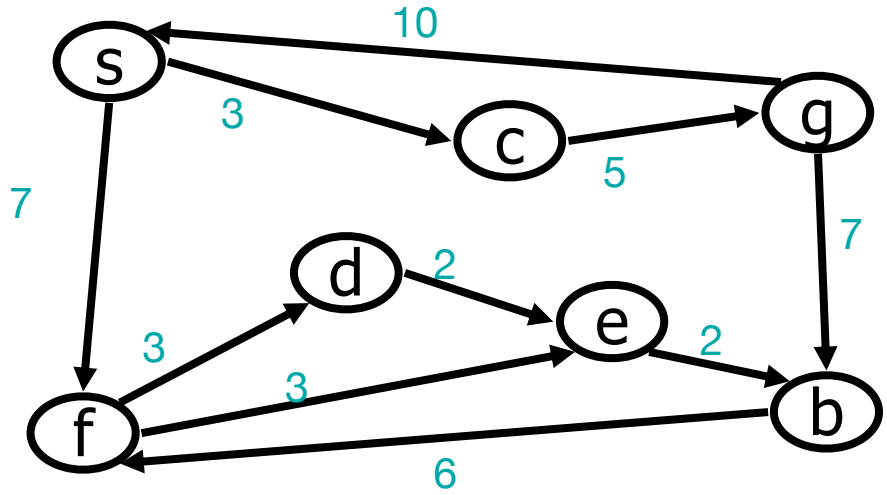


Given a start vertex (source) s, find the path of least total cost from s to every vertex in the graph.

Single Source Shortest Path

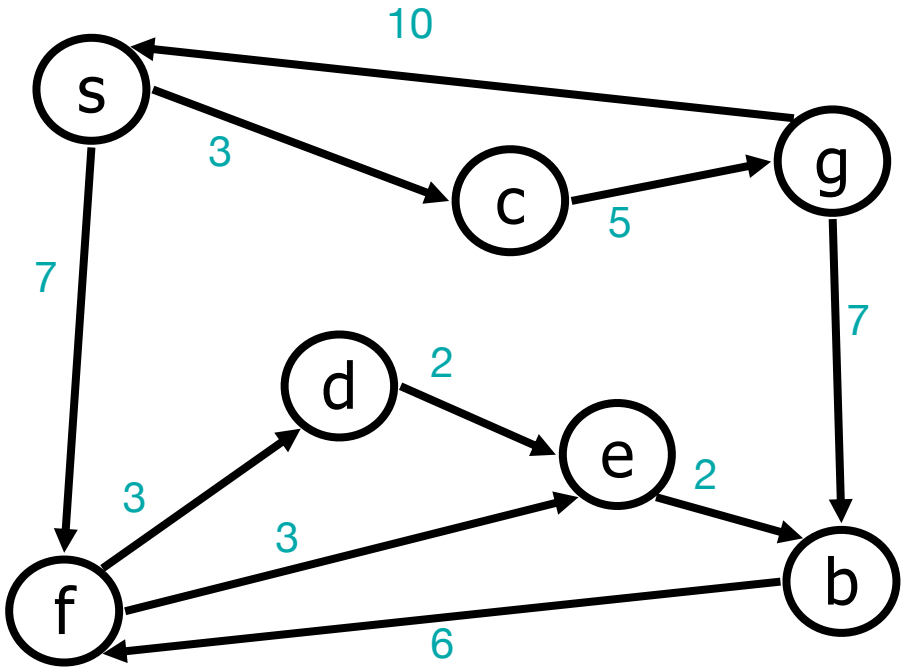
Input: directed graph G with non-negative edge weights, and a start vertex s.

Output: A subgraph G' consisting of the shortest (minimum total cost) paths from s to every other vertex in the graph.



Dijkstra's Algorithm (1959)

Single Source Shortest Path



Given a source vertex s, we wish to find the shortest path from s to every other vertex in the graph.

Initialize structure:

Repeat these steps:

1. Label a new (unlabelled) vertex v, whose shortest distance has been found
2. Update v's neighbors with an improved distance

Single Source Shortest Path

Initialize structure:

1. For all v, d[v] = “infinity”, p[v] = null
2. Initialize source: d[s] = 0
3. Initialize priority (min) queue

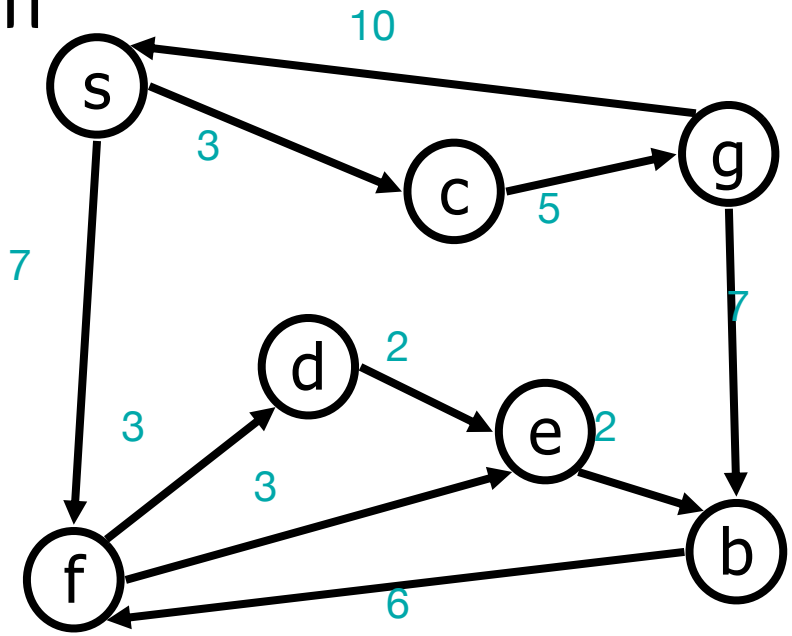
Repeat these steps n times:

- Find minimum d[] unlabelled vertex: v
- Label vertex v
- For all unlabelled neighbors w of v,

If (_____ < d[w])

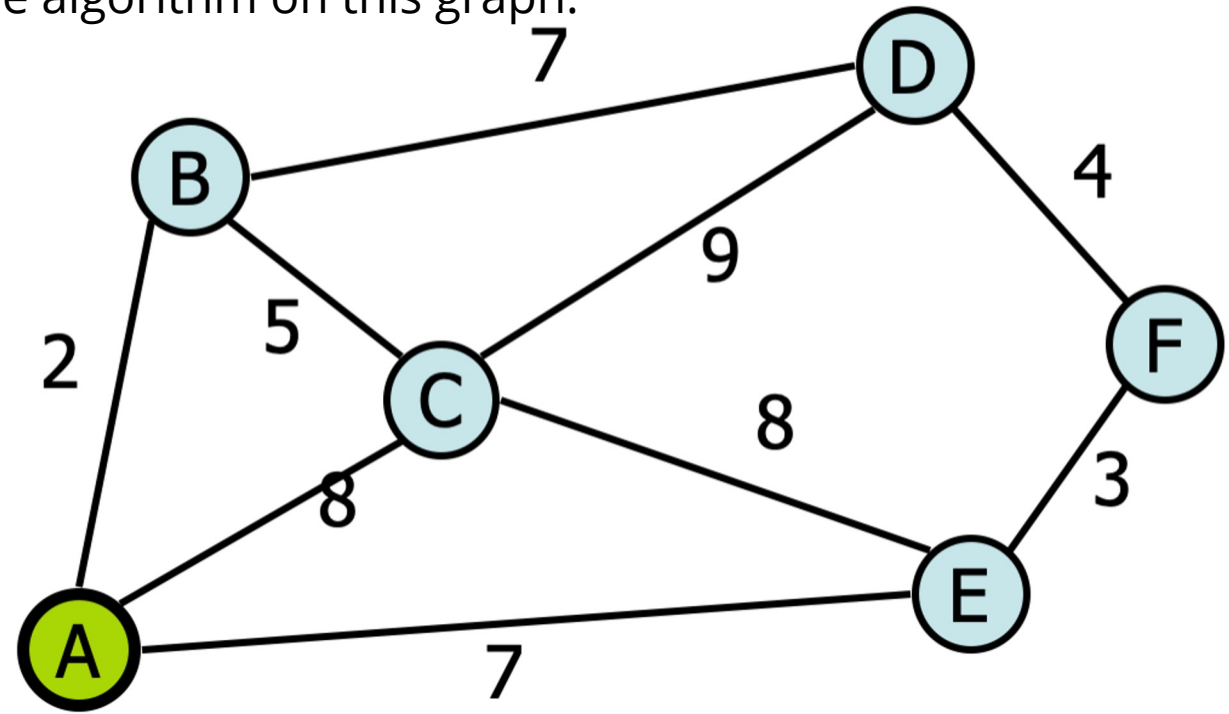
d[w] = _____

p[w] = v



Your Turn...

Execute the algorithm on this graph:



Dijkstra's Algorithm

How is this algorithm similar to BFS/DFS?

_____

_____

How is this algorithm different from BFS/DFS?

_____

_____

Initialize structure:

1. For all v, d[v] = "infinity", p[v] = null
2. Initialize source: d[s] = 0
3. Initialize priority (min) queue
4. Initialize set of labeled vertices to ∅.

Repeat these steps n times:

- Find & remove minimum d[] unlabelled vertex: v
- Label vertex v
- For all unlabelled neighbors w of v,

  If cost(v,w) < d[w]

  d[w] = cost(v,w)

  p[w] = v

Resources...

REALLY great example: https://www.youtube.com/watch?v=wsSEKm-rU6U

OSMNX reference: https://github.com/gboeing/osmnx-examples/tree/master/notebooks

Tutorial: https://gist.github.com/psychemedia/b49c49da365666ba9199d2e27d002d07