Algorithms and Data Structures for Data Science
Weights, Directions, and MST

CS 277
Brad Solomon

November 15, 2021
No lab this week

Last lab will be after break

Remember: 13 labs total, only 12 graded

13th lab will replace lowest grade
Exam 3

Date: December 3rd-5th

Will cover all content prior to fall break
Learning Objectives

Discuss how implementations change due to weight and direction

Discuss how shortest path can be solved w/ weights

Discuss how minimum spanning tree can be solved w/ weights
Graph Implementation: Edge List  \(|V| = n, |E| = m\)

**Vertex Storage:**

**Edge Storage:**
Graph Implementation: Adjacency Matrix

Vertex Storage:

Edge Storage:
**Adjacency List**

```
class vertObj():
    def __init__(self, v):
        self.val = v
        self.eList = []

class edgeObj():
    def __init__(self, e):
        self.v2 = e
        self.parallel = None
```

Weighted / directed graph terminology

- **In-degree:**
- **Out-degree:**
- **Strongly-connected:**
Weighted / directed graph terminology

Which of the following nodes are strongly connected?
Shortest Path
Dijkstra’s Algorithm (SSSP)
Dijkstra’s Algorithm (SSSP)

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

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

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

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

What is the run-time of Dijkstra's?
Landmark Path Problem

What is the shortest path from A to G that visits L?
Minimum Spanning Tree Algorithms

**Input:** Connected, undirected graph $G$ with edge weights (unconstrained, but must be additive)

**Output:** A graph $G'$ with the following properties:
- $G'$ is a **spanning graph** of $G$
- $G'$ is a tree (connected, acyclic)
- $G'$ has a minimal total weight among all spanning trees