CS202 - Algorithm Analysis

Graph Algorithms - Module3

Aravind Mohan

Allegheny College

May 12, 2021

Discussion Based On ...

Sedgewick 4.4

Shortest Path Problem

We are given a starting node **s** $\in$ V and a weighted graph G(V, E, W).

- a node set **V**
- an edge set **E**
- a weight set **W** specifying weights c_{ij} for the edges (i, j) $\in$ E

Problem Definition: The shortest path problem is the problem of determining the shortest path from node **s** to all the other nodes in the graph.

Shortest Path Algorithms

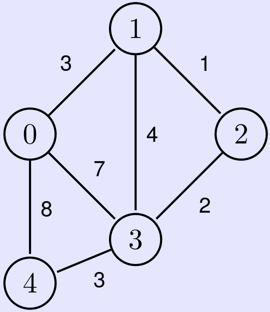
- Dijkstra's algorithm: Solves only the problems with nonnegative costs, i.e., $c_{ij} \geq 0$ for all $(i, j) \in E$
- Bellman-Ford algorithm: Applicable to problems with arbitrary costs
- Floyd-Warshall algorithm: Applicable to problems with arbitrary costs and solves a more general all-to-all shortest path problem

Floyd-Warshall and **Bellman-Ford** algorithm solve the problems on graphs that do not have a cycle with negative cost.

The Power of Dijkstra's algorithm:

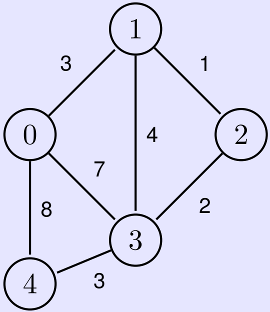
- Find directions between physical locations, such as driving directions on websites like Google Maps or Mapquest
- In data network routing: find the path for data packets to go through a switching network with minimal delay
- Other shortest path problems arising in plant and facility layout, robotics, transportation, and VLSI design

Dijkstra's example



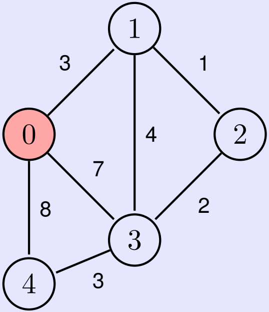
Find shortest path in graph starting from 0

Dijkstra's example



Distance	0	∞	∞	∞	∞
Previous	-1	-1	-1	-1	-1

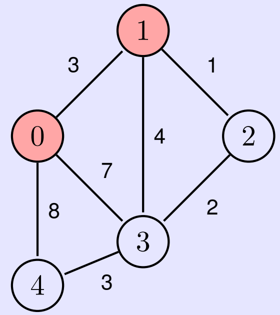
Dijkstra’s example



Distance	~~0~~	3	∞	7	8

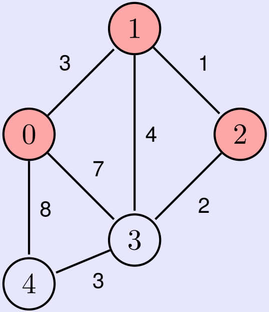
Previous	-1	0	-1	0	0

Dijkstra's example



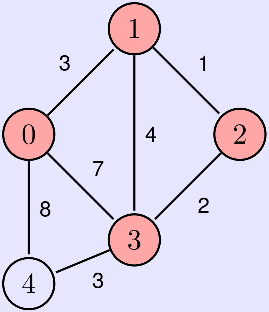
Distance	~~0~~	~~3~~	4	7	8
Previous	-1	0	1	0	0

Dijkstra's example



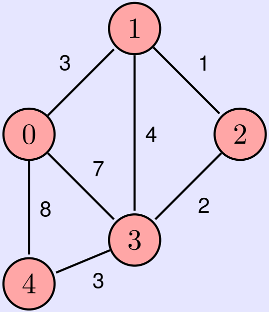
Distance	~~0~~	~~3~~	~~4~~	6	8
Previous	-1	0	1	2	0

Dijkstra's example



Distance	~~0~~	~~3~~	~~4~~	~~6~~	8
Previous	-1	0	1	2	0

Dijkstra's example



Distance	~~0~~	~~3~~	~~4~~	~~6~~	~~8~~

Previous	-1	0	1	2	0

Dijkstra's Algorithm

Graph(G, s)

Input: Graph G = (V, E) directed or undirected, source vertex s $\in$ V

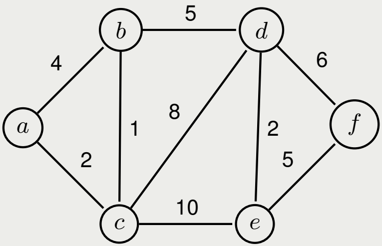
Output: Shortest distance from s to all other vertices in V

```
Let D[s] from s = 0
Let D[t] = ∞ for all vertex t ∈ V − {s}
Let P[t] = undefined for all vertex t ∈ V
while (visited.size <> |V|)
  α = unvisited vertex with minimum distance
  for all u ∈ unvisited neighbors of α
      β = D[α] + E(α, u)
      if (β < D[u])
        D[u] = β
        P[u] = α
      end if
  end for
  visited.add(α)
end while
```

$\boxed{\textbf{Time Complexity - } O(V^2)}$

Try out 1

- **Compute** the shortest path to all vertices from start vertex **a**. Show the Distance and Previous array at every step in your solution.



Interested to learn more Graph Algorithms?

- **Prims Algorithm:**

  https://www.youtube.com/watch?v=A_W4FGPMfDw&list=PLKsSK2k9kZ8yomg0hkI10Jp5PhcAe5qvL&index=2

- **Kruskals Algorithm:**

  https://www.youtube.com/watch?v=kWVncbEm4g0&list=PLKsSK2k9kZ8yomg0hkI10Jp5PhcAe5qvL&index=3

Reading Assignment

Sedgewick 4.4

Questions?

Please ask if there are any Questions
through Slack, Email, and/or during the virtual office hours!