Recitation 9 - Dijkstra's Algorithm

Sebastian Laudenschlager

sebastian.laudenschlager@colorado.edu

April 6, 2018

Dijktra's algorithm

- What is it?
  - Single Source Shortest Path (SSSP) algorithm.
  - Given a source vertex, Dijkstra's algorithm will find the shortest path to every other vertex in a graph.
- Essentially a weighted version of BFS.
  - Requires nonnegative edge weights.

Dijkstra details

- Dijkstra's algorithm maintains a set S containing vertices whose final shortest-path from the source has already been computed.
- Repeatedly select vertices $u \in V - S$ with minimum shortest-path estimate and adds u to S, then relaxes all edges leaving u.
- Use a min priority queue (e.g. min-heap) based on shortest-path weights.

Pseudocode

```
def dijkstra(G, w, s):
    initSingleSource(G, s)
    S = {}
    Q = G.V
    while Q not empty:
        u = extractMin(Q)
        S = S + {u}
        for each vertex v in G.adj[u]:
            relax(u, v, w)
```

Edge relaxation

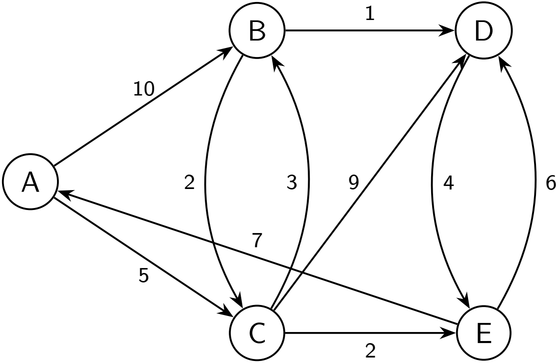
- **Relaxing** an edge (u, v) is the process of testing whether we can improve our shortest path estimate by going through u.
- If so, update $v.d$ and $v.pi$ accordingly.

```
def relax(u, v, w):
    if v.d > u.d + w(u, v):
        v.d = u.d + w(u, v)
        v.pi = u
```

- If the current shortest path estimate of getting to v is greater than the estimate to get to u plus the cost of going from u to v, then we've found a shorter path to v through u.
- So update our estimates and predecessor information accordingly.

Example

- Consider the following directed graph:



- Goal: Run Dijkstra's algorithm to find the shortest paths from A to all other vertices.

Running Time

- Intuitive derivation of running time:
  - In Dijkstra's algorithm, we explore every vertex and every edge.
  - For each such explorations, we perform a `extractMin` operation, which takes $\mathcal{O}(\log V)$ time.
  - Thus, we have a running time of $\mathcal{O}((V + E) \log V)$.
  - If we have a connected graph, i.e. $V = \mathcal{O}(E)$, then this becomes $\mathcal{O}(E \log V)$.

Cycles?

- Shortest paths cannot contain cycles. Why?
  - Already ruled out negative weight cycles.
  - Can't have positive weight cycles either, because we could just get a shorter path by omitting the cycle.
  - 0 weight cycles can also be omitted entirely; there is no need to use them.

Applications

- Robot path finding on a grid.
- GPS navigation.
- Updating network router forwarding tables.