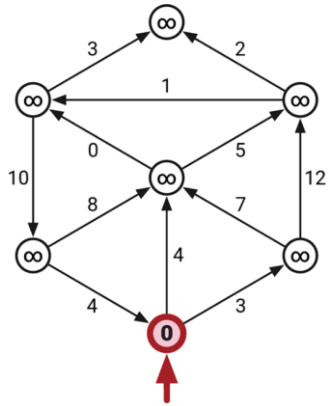
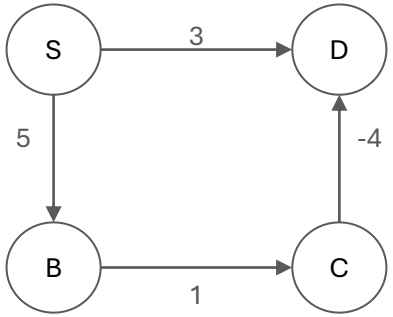
Algorithm Design & Analysis (CSE222)

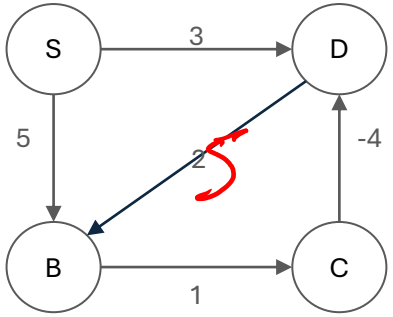
Lecture-19

Recap

- Shortest Path
  - Dijkstra's algorithm (single source)








```
DIJKSTRA(s):
  INITSSSP(s)
  INSERT(s, 0)
  while the priority queue is not empty
      u ← EXTRACTMIN( )
      for all edges u→v
          if u→v is tense
              RELAX(u→v)
              if v is in the priority queue
                  DECREASEKEY(v, dist(v))
              else
                  INSERT(v, dist(v))
```

Outline

- Shortest Path
  - Bellman Ford
- Distance Vector Protocols
- Network Flows

Dijkstra’s Algorithm

When there is a negative edge, Dijkstra’s algorithms may or may not return correct answer.

- In the presence of negative cycle, shortest paths are undefined.

- In the presence of negative edge but not a negative cycle then Dijkstra returns correct answer.
  - How to ensure correct answers?

Bellman Ford

Claim: If G has no negative cycles, then there is a shortest path from s to t that is simple (i.e., does not repeat nodes), and hence has at most n-1 edges.

Proof:

- Since, every cycle (if present) has a nonnegative cost then no path from s to t would not repeat any vertex v.
- If a vertex repeats in a path then we can remove the portion of the path. This will not increase the path cost from s to t.
- So at max n-1 edges are traversed to reach t from s.

Using dynamic programming by taking advantage of the above claim.

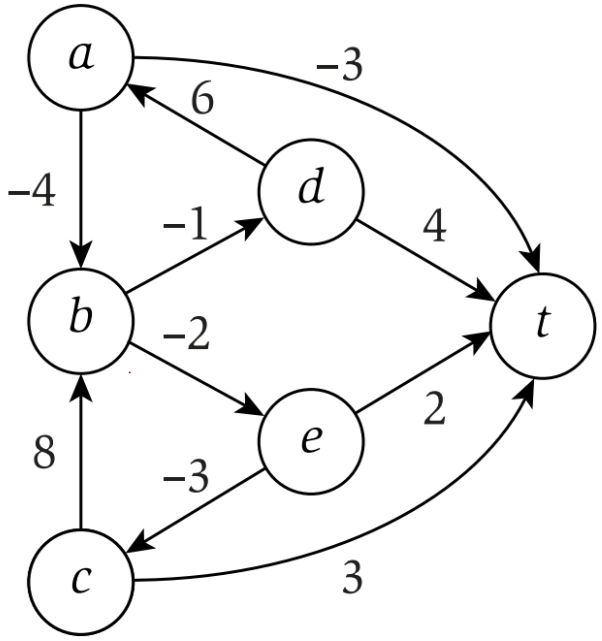
Bellman Ford

Using dynamic programming.
Subproblem: opt(i, v) is the minimum cost of v-t path using at most i edges.

- If path uses at most i-1 edge then opt(i, v) = opt(i-1, v)
- If path uses i edges and there is an edge (v, w) then opt(i, v) = opt(i-1,w) + c_{vw}

opt(i, v) = $\min\{OPT(i-1, v), \min_{w \in V}\{OPT(i-1, w) + c_{vw}\}\}$

Bellman Ford



opt	0	1	2	3	4	5
t	0	0	0	0	0	0
a	inf	-3	-3	-4	-6	-6
b	inf	∞	0	-2	-2	-2
c	inf	3	3	3	3	3
d	inf	4	3	3	2	0
e	inf	2	0	0	0	0

Bellman Ford

Running time: Depends on the table size and time compute value at every index.
$O(n^3)$

Can we improve its running time?

$$OPT(i,v) = \min\left(OPT(i-1,v),\ \min_{w \in N_v} OPT(i-1,w) + c_{vw}\right)$$

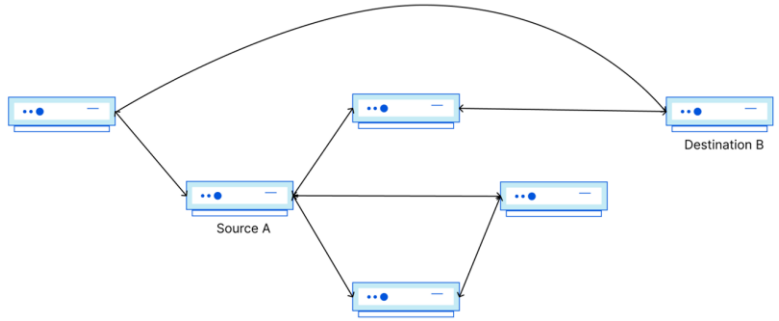
$$n \cdot \sum_{v \in V} N_v = O(n*|E|)$$

Outline

- Shortest Path
  - Bellman Ford
- Distance Vector Protocols
- Network Flows

Application: Distance Vector Protocols

- Nodes are routers.
- Edges are direct links between routers.
- Edge weights are the cost represents dela
  the link.



Problem: Find a path from s to t with a minimum delay.
Delays are non-negative values. So, we can use Dijkstra.
In case of large network global knowledge is difficult.

Use Bellman Ford algorithm.

Problems

Design an algorithm that detects if a graph contains negative cycles.

Outline

- Shortest Path
  - Bellman Ford
- Distance Vector Protocols
- Network Flows

Network Flow

An (s, t)-flow is a function f: E → R that satisfies the following conservation constraints for every vertex v (possibly) except s and t.

$$\sum_u f(u \to v) = \sum_w f(v \to w)$$

Total flow into v is equal to the total flow out of v.

Flow is defined as |f|. It is the total flow out of the source s.

$$|f| := \sum_w f(s \to w) - \sum_u f(u \to s)$$

Prove that the net flow into t (the sink) is |f|.

Capacity is a function c: E → $R_{\geq 0}$.

Network Flow

An (s, t)-flow is a function f: E → R that satisfies the following conservation constraints for every vertex v (possibly) except s and t.

$$\sum_u f(u \to v) = \sum_w f(v \to w)$$

Capacity is a function c: E → $R_{\geq 0}$.

- A flow f is feasible if f(e) ≤ c(e) for every edge e in G.
- A flow f saturates an edge e if f(e) = c(e).
- A flow f avoids an edge e if f(e) = 0.

Problem: Given a graph with capacity find a maximum feasible flow.

Cut

An (s, t)-cut is a partition of vertices into disjoint set S and T, where s ∈ S and t ∈ T

The capacity of a cut is defined as

$$\|S, T\| := \sum_{v \in S} \sum_{w \in T} c(v \to w)$$

Problem: Given a graph with capacity find the minimum cut.

Reference

Slides

Jeff Erickson Chp-10

Algorithms Design by Kleinberg & Tardos - Chp 6.8 - 6.10