Bellman-Ford continues
Minimum Spanning Tree
Bellman - Ford

dist = array indexed by V initialized to ∞
prec = array indexed by V initialized to 1

dist[S] = 0

for l = 1 to |V| - 1:
   for each v in V:
      for each edge (u, v):
         if dist[u] + l(u, v) < dist[v]:
            dist[v] = dist[u] + l(u, v)
            prec[v] = u

Running time \( O(|V| \cdot |E|) \)

Correctness: At step \( l \) of outer for
for every v

\[ \text{dist}[v] \leq \text{length of shortest path from s to v that uses } \leq l \text{ edges} \]
Bellman-Ford

dist = array indexed by V initialized to ∞
prec = array indexed by V initialized to ⊥

def update (u, v):
    if dist[u] + e(u, v) < dist[v]:
        dist[v] = dist[u] + e(u, v)
        prec[v] = u

dist[s] = 0

for t = 1 to |V| - 1:
    for each (u, v) in E:
        update (u, v)
For every \( t = 1, \ldots \), after \( t \) iterations of outer "for" loop:

\[
dist[v] \leq \text{length of shortest path from } s \text{ to } v \text{ that uses } \leq t \text{ edges}
\]

- \( t = 0 \)
- \( dist[s] = 0 \)
- \( dist[v] = \infty \text{ for } v \neq 0 \)

Assume true up to \( t \):

Consider \( dist[v] \) after \( t+1 \) executions of "for" loop

\( \forall v \)

\( \forall \) path \( P \) from \( s \) to \( v \) that uses \( \leq t+1 \) edges

want to prove: \( dist[v] \leq \text{length } P \)

\[
\text{length of } P = \underbrace{l(s,v_1) + l(v_1,v_2) + \cdots + l(v_{t-1}, v_t)}_{\text{dist } [v_6] \leq 5} + l(v_t, v)
\]
∀ ν

∀ path P from s to ν that uses ≤ t+1 edges

want to prove: dist[ν] ≤ length P

consider path P from s to ν with = t+1 edges

length of P = \( l(s, v_1) + l(v_1, v_2) + \cdots + l(v_{t-1}, v_t) + l(v_t, ν) \)

\[ \text{dist}[ν] \leq \frac{l(s, v_1) + l(v_1, v_2) + \cdots + l(v_{t-1}, v_t) + l(v_t, ν)}{t} \]

at end of iteration t

After update(ν, v) in iteration t+1

\[ \text{dist}[ν] \leq \text{dist}[ν_t] + l(ν_t, v) \leq \text{length P} \]

At end of iteration t+1

\[ \text{dist}[ν] \leq \text{length P} \]
After 0 iterations
\[ \text{dist} [sJ] = 0 \]

After 1 iteration
consider step in which update \((s, A)\)
in 1st iteration

\[ \text{dist} [AJ] \leq \ell (s, A) \text{ remains true in all subsequent steps} \]

After 2nd iteration
consider step in which update \((A, B)\)

\[ \text{dist} [BJ] \leq \text{dist} [AJ] + \ell (A, B) \]
\[ \leq \ell (s, A) + \ell (A, B) \]

After 3rd iteration
\[ \text{update} (B, C) \]

\[ \text{dist} [CJ] \leq \text{dist} (BJ) + \ell (B, C) \]
\[ \leq \ell (s, A) + \ell (A, B) + \ell (B, C) \]

After 4th iteration
\[ \text{dist} [VJ] \leq \ell (s, A) + \ell (A, B) \]
\[ + \ell (B, C) + \ell (C, V) \]
Tree
Graph: undirected, connected, acyclic

Note: undirected a cycle is of length \( \geq 3 \)

directed

\[ A \rightarrow B \]

undirected

\[ A \overset{\circ}{\leftrightarrow} B \]

Tree on \( n \) vertices has \( = n - 1 \) edges

A graph is a tree \( \iff \) it is connected and \( 1 \text{V}-1 \) edges
Suppose $G$ is connected and has a cycle. Then we can remove any one edge from the cycle without compromising connectivity.
Thm: Suppose $G$ is connected, all edge costs are different, $S \subseteq V$, $1 \leq |S| \leq |V|-1$, then cheapest edge out of $S$ must belong to all minimum spanning trees.

Proof:

Let $S$ be a set of vertices, $(u,v)$ be cheapest edge out of $S$.

Suppose $T$ is an optimal tree that does not use $(u,v)$.

- Add $(u,v)$ to $T$.
  We create a cycle.
  Let $(z,w)$ be a $T$-edge in the cycle that crosses from $S$ to $V-S$.

- Take out $z,w$.
  We a new tree $(T \cup \{(u,v)\}) - \{z,w\}$ of cost $\text{cost}(T) + \text{cost}(u,v) - \text{cost}(z,w) < \text{cost}(T)$.

$T$ is not optimal.
Let $G$ be an undirected graph. Let $S$ be a subset of vertices and $(u,v)$ be the cheapest edge out of $S$. Then there is an optimal tree that contains $(u,v)$. 

![Diagram of a graph with vertices labeled 0, 1, 2, and 3, and edges connecting them with weights 1, 2, and 1. The vertex labeled 2 is connected to vertices 0 and 3, and the vertex labeled 1 is connected to vertex 3.]
Thm
G connected weighted graph
S subset of vertices
F set of edges of G that
- don't create any cycle
- don't cross from S to V-S
(u,v) a cheapest edge from s to V-S

Then size of minimum spanning tree of G including all edges of F
size of minimum spanning tree of G including all edges of F and also (u,v)