18.2.3
The Bellman-Ford Algorithm

Bellman-Ford Algorithm

```
for each u ∈ V do
    d(u, 0) ← ∞
d(s, 0) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        d(v, k) ← d(v, k − 1)
        for each edge (u, v) ∈ in(v) do
            d(v, k) = min{d(v, k), d(u, k − 1) + ℓ(u, v)}

for each v ∈ V do
    dist(s, v) ← d(v, n − 1)
```

Running time: $O(mn)$ Space: $O(m + n^2)$

Space can be reduced to $O(m + n)$.

Bellman-Ford Algorithm

```
for each u ∈ V do
    d(u, 0) ← ∞
d(s, 0) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        d(v, k) ← d(v, k − 1)
        for each edge (u, v) ∈ in(v) do
            d(v, k) = min{d(v, k), d(u, k − 1) + ℓ(u, v)}

for each v ∈ V do
    dist(s, v) ← d(v, n − 1)
```

Running time: $O(mn)$ Space: $O(m + n^2)$

Space can be reduced to $O(m + n)$.

Bellman-Ford Algorithm

```
for each u ∈ V do
    d(u, 0) ← ∞
d(s, 0) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        d(v, k) ← d(v, k − 1)
        for each edge (u, v) ∈ in(v) do
            d(v, k) = min{d(v, k), d(u, k − 1) + ℓ(u, v)}

for each v ∈ V do
    dist(s, v) ← d(v, n − 1)
```

Running time: $O(mn)$ Space: $O(m + n^2)$

Space can be reduced to $O(m + n)$.

Bellman-Ford Algorithm

```
for each u ∈ V do
    d(u, 0) ← ∞
d(s, 0) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        d(v, k) ← d(v, k − 1)
        for each edge (u, v) ∈ in(v) do
            d(v, k) = min{d(v, k), d(u, k − 1) + ℓ(u, v)}

for each v ∈ V do
    dist(s, v) ← d(v, n − 1)
```

Running time: $O(mn)$ Space: $O(m + n^2)$

Space can be reduced to $O(m + n)$.

Bellman-Ford Algorithm

```
for each u ∈ V do
    d(u, 0) ← ∞
d(s, 0) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        d(v, k) ← d(v, k − 1)
        for each edge (u, v) ∈ in(v) do
            d(v, k) = min{d(v, k), d(u, k − 1) + ℓ(u, v)}

for each v ∈ V do
    dist(s, v) ← d(v, n − 1)
```

Running time: $O(mn)$ Space: $O(m + n^2)$

Space can be reduced to $O(m + n)$.

Bellman-Ford Algorithm

```
for each u ∈ V do
    d(u, 0) ← ∞
d(s, 0) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        d(v, k) ← d(v, k − 1)
        for each edge (u, v) ∈ in(v) do
            d(v, k) = min{d(v, k), d(u, k − 1) + ℓ(u, v)}

for each v ∈ V do
    dist(s, v) ← d(v, n − 1)
```

Running time: $O(mn)$ Space: $O(m + n^2)$

Space can be reduced to $O(m + n)$.

Bellman-Ford Algorithm: Cleaner version

```
for each u ∈ V do
    d(u) ← ∞
d(s) ← 0

for k = 1 to n − 1 do
    for each v ∈ V do
        for each edge (u, v) ∈ in(v) do
            d(v) = min{d(v), d(u) + ℓ(u, v)}

for each v ∈ V do
        dist(s, v) ← d(v)
```

Running time: $O(mn)$ Space: $O(m + n)$

Exercise: Argue that this achieves same results as algorithm on previous slide.

THE END

...

(for now)