1 Fundamental Shortest Paths Formula

For any vertex w that isn't the source, $w \neq s$,

$$\text{dist}(w) = \min_{(v,w)\in E} \text{dist}(v) + \ell(v, w)$$

We can use a system of equations to solve for the distances. When $\ell \geq 0$, Dijsktra gives an order to solve in.

1.1 Algorithm Idea

Instead of finding the shortest paths, which may not exist, we instead find the shortest paths of length at most k. So, for $w \neq s$, we have:

$$\text{dist}_k(w) = \min_{(v,w)\in E} \text{dist}_{k-1}(v) + \ell(v, w)$$

1.2 Algorithm

```
Bellman-Ford(G, s, l)
    dist_{0}(v) = infinity for all v
    dist_{0}(s) = 0
    For k = 1 to n
        For w in V
            dist_{k}(w) = min(dist_{k - 1}(v) + l(v, w))
        dist_{k}(s) = min(dist_{k}(s), 0)
```

1.3 Analysis

Proposition. *If $n \geq |V| - 1$ and if G has no negative weight cycles, then for all v,*

$$dist(v) = dist_n(v)$$

In particular:

- If there is a negative weight cycle, there is probably no shortest path.
- If not, we only need to run our algorithm for $|V|$ rounds, for a final runtime of $\mathcal{O}(|V||E|)$.

1.4 Detecting Negative Cycles

If there are no negative weight cycles, Bellman-Ford computes shortest paths (and they might not exist otherwise). Howe do we know whether or not there are any?

1.4.1 Cycle Detection

Proposition. *For any $n \geq |V| - 1$, there are no negative weight cycles reachable from s if and only if, for every $v \in V$:*

$$dist_n(v) = dist_{n+1}(v)$$

1.5 Potential Function

Let

$$\ell'(v, w) = \ell(v, w) - d(v) + d(w) \geq 0$$

Imagine someone lending you $d(w)$ when you arrive at w, but then you have to pay it back when you leave.