BF SSSP algorithm

Repeat [VI-1 times
"Contract"

Relax \((u, v)\)

If \(\delta(u) > \delta(v) + w(u, v)\)
- then \(\delta(u) := \delta(u) + w(u, v)\)

Relax every edge exactly once in any order

Running time: \(O(V(VI + E))\)

If we have a shortest path to \(v\)
\(s \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \cdots v_k \rightarrow \bar{v} \rightarrow v\)
then any subpath is also shortest

Observation on relaxation: If \(\delta(v_j) = \delta(v_j)\) an upper bound on the exact distance
- then the next relaxation of \((\bar{v}, v)\)
will make \(\delta(v) = \Delta(v)\)
Claim: All vertices has i hops from s will get the correct shortest path label within i iterations.

\[ \Rightarrow \text{ all vertices will have correct label within } (V_1 - 1) \text{ hops.} \]

No negative cycle.

If there is a negative cycle, the algorithm reports it.

Downside: \( O(|V_1|E_1) \) is too expensive.

\( O(n^3) \)

Observation: If all weights are non-negative, then along a shortest path, \( \Delta(v_j) \) \( \Delta(v_{j-1}) \) \( \Delta(v_{j-2}) \) \( \Delta(3) = 0 \)
Dijkstra: In increasing shortest path distance (not hops)

In Dijkstra we know when we have achieved $\delta(v) = \Delta(v)$

How many times do we relax an edge in BF? $O(1V1)$

Dijkstra: $O(1)$

Claim: In any round, $\delta(v) = \Delta(v)$ for the vertex with max $\delta$

Relaxing an edge: Decreasing labels: heat
Proof (by contradiction): Suppose the vertex \( v^* \) that we chose doesn't satisfy \( \delta(v^*) = \Delta(v^*) \).

Clearly \( v_j \) doesn't have \( \delta(v_j) = \Delta(v_j) \).

The rigorous proof should look at the first iteration when things went wrong.

Because of the relaxation operation, the algorithms compute the shortest paths in strict ordering from \( s \):
In a DAG, we can define an ordering based on if \( u \rightarrow v \rightarrow u \) no\( u \rightarrow v \rightarrow u \) may exist.

It is a partial order and this can be computed by topological sort.

\[
\text{in } O(V + E) \text{-time}
\]

\[v_1, v_2, v_3, \ldots, v_n\]

**Assignment:** Write out the formal algorithm and prove correctness.

**Running:** (given the ordering) - time analysis.

Single Source Shortest Paths

Computing distance between all pairs: can be done by running SSSP from all sources.

\[BF = \{V\}, \{V\}, |E| = |V|^2 E, O(n^4)\]
Dijkstra: \( O((V^1)(V^1 + (E^1) \log n)) \)

\( O(n^3 \log n) \)

\( \omega(x_i, y) \)
We will define a notion of matrix multiplication in the following manner:

\[
C = A \cdot B, \quad c_{ij} = \min_{1 \leq k \leq n} q_{ik} + b_{kj}
\]

\[
\begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} \\
  a_{21} & \ddots & \ddots & \vdots \\
  \vdots & \ddots & \ddots & a_{2n} \\
  a_{m1} & \cdots & a_{mn}
\end{bmatrix}
\begin{bmatrix}
  b_{11} & b_{12} & \cdots & b_{1n} \\
  b_{21} & \ddots & \ddots & \vdots \\
  \vdots & \ddots & \ddots & b_{2n} \\
  b_{m1} & \cdots & b_{mn}
\end{bmatrix}
\]

\[
\leq a_{i\times j} \cdot b_{j\times \cdot}
\]