CSL 630 lecture 2 — July 28

BF SSSP algorithm

Repeat $|V|-1$ times
"Contract"

Relax every edge exactly once in any order

<u>Relax</u> $(\overrightarrow{u,v})$

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

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

If we have a shortest path to v

$s \to v_1 \to \underbrace{v_2 \to v_3 \cdots v_k}\; v_j \to v$

then any subpath is also shortest

Observation on relaxation : If $\delta(v_j) = \Delta(v_j)$

upperbound — exact distance

then the next relaxation of $(\overrightarrow{v_j, v})$ will make $\delta(v) = \Delta(v)$

Claim: All vertices whose shortest path has i hops hops from s will get the correct shortest path label within i iterations.

$\Rightarrow$ all vertices will have correct label within $|V|-1$ hops.

no-negative cycle

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

Downside: $O(|V|.|E|)$ is too expensive

$O(n^3)$

$$s \to v_1 \to v_2 \to \cdot \to \cdot \to \cdot \to \cdot \to v_j \to v$$

Observation: If all weights are non-negative then along a shortest path

$$\Delta(v) \geq \Delta(v_j) \geq \Delta(v_{j-1}) \ \cdots \ \Delta(s) = 0$$

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: $O(|V|)$

Dijkstra: $O(1)$*

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

heap

- Relaxing an edge: *decreasing labels:

Proof (by contradiction): Suppose the vertex v^* that we chose doesn't satisfy $\delta(v^*) = \Delta(v^*)$ $>$

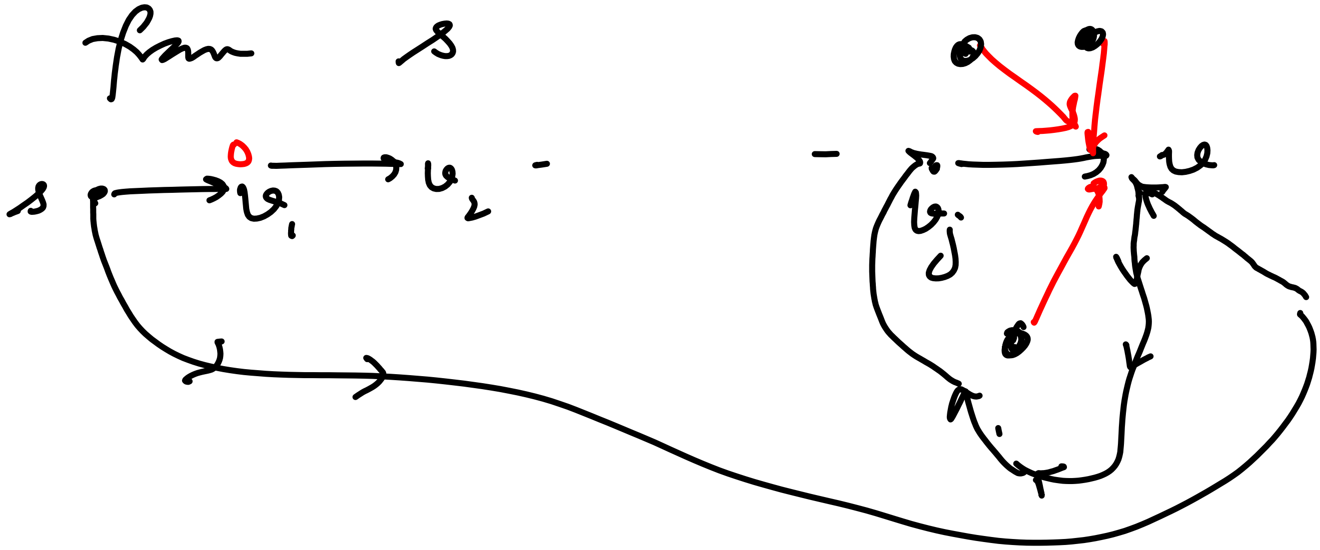
shortest path

$s \longrightarrow v_1 \longrightarrow v_2 \; - \; \boxed{}\, v_l \qquad v_j \longrightarrow v^*$

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 $v_j \rightsquigarrow v$ or $v \rightsquigarrow v_j$.
(none may exist)

It is a partial order and this can be computed by topological sort.
(in $O(|V|+|E|)$-time

$$v_1 \quad v_2 \quad v_3 \quad \cdots \quad v_n$$

Assignment: Write out the formal algorithm and prove correctness ↓ Running-time analysis

Running: (given the ordering)

Single Source Shortest Paths

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

BF: $|V| \cdot |V| \cdot |E| = |V|^2 E$

$O(n^4)$

Dijkstra: $O(|V| \cdot (|V| + |E| \cdot \log n)$

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

$$A: \begin{array}{c|ccccc} & v_1 & v_2 & v_3 & v_j & v_n \\ \hline v_1 & & & & & \\ v_2 & & & & & \\ \vdots & & & & & \\ v_i & & & & \boxed{w(v_i, v_j)} & \\ \vdots & & & & & \\ v_n & & & & & \end{array}$$

We will define a notion of matrix multiplication in the following manner

$$C = A^{n\times n} \cdot B^{n\times n} \qquad c_{ij} = \min_{1 \le k \le n} \{ a_{ik} + b_{kj} \}$$

$$\begin{array}{c c} & \\ & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \cdots & & a_{2n} \\ \cdot & & & \\ & & & \\ a_{n1} & a_{n2} & \cdot & a_{nn} \end{bmatrix} \end{array} \begin{bmatrix} b_{11} & b_{12} & \cdot & b_{1n} \\ b_{21} & & & b_{2n} \\ & & & \\ & & & \\ b_{n1} & & & b_{nn} \end{bmatrix}$$

$$\begin{array}{l} \\ i \end{array}$$

$$\sum a_{ik} \times b_{kj}$$