From the Min-Plus Product to All Pairs Negative Triangle

Construct sets of nodes I, K, J of size n each

Edge \((i,j) \in I \times K \times J\)

Weight of edges from J to I is additive

Goal: \(V_{i,j} = \text{compute } \min_k (w(i,k) + w(k,j))\)

This is a k will

\(= \text{minimum integer } z \text{ s.t. } w(i,k) + w(k,j) < z + 1\)

Idea: For \(k \geq i\) For every \(i,j\), we can test whether

\[ w(i,k) + w(k,j) < z(i,j) + 1 \]

\(\iff\)

\[ w(i,k) + w(k,j) + (-z(i,j) - 1) < 0 \]

by setting \(w(i,j) := -z(i,j) - 1\) and checking if there is a neg. triangle containing \(i\) and \(j\)

\(\implies\) Find \(\min\) such \(z(i,j)\) by simultaneous binary search

Weights are in \([-n^2, \ldots, n^2]\)

Each entry of Min-Plus Product is in \([-2n^2, \ldots, 2n^2]\)

Binary search takes \(\log (4n^2 + 1) = O(\log n)\) steps

\(T(n) = \text{time alg. for All-Pairs Min } \Rightarrow T(n) \text{ high-time for Min-Plus}

Algorithm: \(V_{i,j} = \text{initialize } \hat{M}_{i,j} = -2n^2 \text{ and } M[i,j] = 2n^2\)

Repeat \(\log (4n^2)\) times:

\[ V[i,j] := w(i,j) := -\hat{M}[i,j] + M[i,j] / 2 \]

Compute All-Pairs Neg. Triangle

\[ V[i,j] := \text{if } (i,j) \text{ is in neg. triangle: } M[i,j] := w(i,j) - 1 \]

\[ \text{otherwise: } m[i,j] := -w[j,i] \]

\[ V[i,j] = M[i,j] \]
The reduction Min-Plus Product $\rightarrow$ APSP is almost trivial:

Construct graph $G$:

$$w(i, k) = A[i, k]$$
$$w(k, j) = B[k, j]$$

Let $u$ be $i$-th node on the left, let $v$ be $j$-th node on the right.

Then the shortest path from $u$ to $v$ has weight

$$\min_k (w(i, k) + w(k, j)) = \min_k (A[i, k] + B[k, j]) = C[i, j]$$

**Theorem:** Let $A$ be the weighted adjacency matrix of a graph $G$ with nodes $V = \{v_1, \ldots, v_n\}$ and let $B$ be the same as $A$ with every diagonal entry set to 0, i.e. $B = A \oplus I$ in the min-plus semiring.

Then, the matrix $C := \{B^k\}_{k=0}^{n}$ (with exponentiation according to min-plus product) contains, for every $i, j$, entry $C[i, j] = \text{length of shortest path} \leq k$ from $v_i$ to $v_j$.

**Proof:** We prove by induction on $k$.

For all $i, j$,

$$B^0[i, j] = \text{dist}^0[i, j], \text{ i.e. length of shortest path from } v_i \text{ to } v_j \text{ with } \leq 0 \text{ edges}$$

Base case $k = 0$: $B^0 = I$ (every node at distance 0 from itself)

Inductive step: $k \rightarrow k + 1$: $B^{k+1} = B^k \circ B = B^k \circ (A \oplus I)$

$$= (B^k \circ A) \oplus B^k \circ D = (B^k \circ A) \oplus B^k$$

By IH: $B^k$ contains lengths of shortest paths with $\leq k$ edges.

Every shortest path with $\leq k + 1$ edges consists of $\leq k$ edges or first $k$ shortest paths with $\leq k$ edges and then a single edge.
The theorem now follows because

Thus: \( \text{dist}^{k+1}(v_i, v_j) = \min_k \left( \min_{k=1}^{k+1} \left( \text{dist}^k(v_i, v_k) + \text{dist}^k(v_k, v_j) \right) \right) \)

\( = \min \left( (B^k \otimes A)[i, j], B^{k+1}[i, j] \right) \)

\( = B^{k+1}[i, j] \)

The theorem now follows because the shortest path from \( v_i \) to \( v_j \) (for all \( i, j \)) has at most \( n \) edges and

\( n-1 \leq 2 \text{log}_2 n \)

\( \square \)

**How to compute \( B^{2 \text{log}_2 n} \)?**

Repeated squaring:

\[
B = B_0, B_0^2 = B_1, B_1^2 = B_2, \ldots
\]

\[
B^t = B_0^t \otimes B_0^t \otimes \ldots \otimes B_0^t \]

\( t = \log_2 (2 \text{log}_2 n) = O(\log n) \)

**Then:** If \( \text{Min-Plus Product} \) has a \( T(n) \)-time algorithm,

then \( \text{APSP} \) has an \( O(T(n) + n^2) \log n \)-time algorithm.

**Remark:** The log \( n \) factor can be eliminated

(see [Lipton/Repscroft/Ullman 1974])

**Remark:** In the APSP problem, non-edges, i.e., pairs of nodes \( u, v \) such \( (u, v) \notin E \), can be simulated by:

(a) adding an edge \( (u, v) \) of weight \( \infty \) to the graph.

(b) adding an edge \( (u, v) \) of weight \( nW \) to the graph (where \( W \) is the maximum weight in the input graph).

**Reason for b:** shortest path has length \( \leq (n-1) \cdot nW \)

If shortest path after adding edges in step (b) has length \( > (n-1)W \), we know that initially no path existed.
**Sublinear Equivalence of Radius**

**Definition (Radius):**

**Input:** Weighted directed graph $G$ with edge weights $c \in \{0, 1, \ldots, M\}$

**Task:** Output $\min \max_{u \neq v} \text{dist}(u,v)$ for some constant $c$

where $\text{dist}(u,v)$: length of shortest path from $u$ to $v$

**Intuition:** $u$ is the most central vertex

We will prove:

**Thm:** APSP $\equiv$ Radius (sublinear equivalent)

by showing

**Neg. Triangle $\leq$ Radius and Radius $\leq$ APSP**

**Reduction: Radius $\rightarrow$ APSP**

- Compute all pairwise distances $T_{\text{APSP}}(n)$
- Evaluate definition of radius $O(n^2)$

$\rightarrow$ sublinear reduction with one oracle call

**Reduction: Negative Triangle $\rightarrow$ Radius**

Given directed graph with $n$ nodes $\{1, \ldots, n\}$ and edge weights in $\{-M, -M+1, \ldots, M\}$ where $M = n^c$ for some constant $c$

Construct directed graph $H$ with $O(n)$ nodes and edge weights in $\{0, 1, \ldots, O(M)\}$

1. Make 4 layers $\{A, B, C, D\}$ with $n$ nodes each $\mathcal{A} = \{i_A, \ldots, n_A\}$, etc.
2. For every edge $(i, j)$ of $G$: Add $(i_A, j_B)$, $(i_B, j_C)$, $(i_C, j_D)$ of weight $M \cdot (i,j) / 4$
3. For every $i_A \neq h$, add edges of weight $3M-1$ from $i_A$ to all other nodes except $i_D$ and $i_A$
**Lemma:** The radius of $H$ is $\leq 3M - 1$ if and only if $G$ contains a negative triangle if and only if the radius of $H$ is $\leq 3M - 1$

**Proof:**

\(\Rightarrow\) Consider triangle \(\vec{i} \xrightarrow{j} \vec{k} \xrightarrow{j'} \vec{i} \) of total weight $W \leq -1$

Then in $H$ there is a path \(i_A \to j_B \to k_C \to i_D\) of total weight $3M + W \leq 3M - 1$ (by rule (2))

\(\Rightarrow\) dist \((i_A, i_D) \leq 3M - 1\)

For any other vertex \(i \neq i_A, i_D\), dist \((i_A, i_D) \leq 3M - 1\) by (3)

\(\Rightarrow\) max \(\text{dist}(i_A, i) \leq 3M - 1\)

\(\Rightarrow\) max \(\min_m \text{max \dist}(i_A, i) \leq 3M - 1\)

\(\Leftarrow\) We have \(\min_m \text{max \dist}(u, v) \leq 3M - 1\)

If \(u \neq i_A\) for some \(1 \leq i \leq v\), then

\(\text{dist}(u, i_A) = \infty\) for any \(i \neq A\)

\(\Rightarrow\) max \(\text{dist}(u, v) = \infty\)

Thus \(\min_m \text{max \dist}(u, v)\) is minimized by some \(u = i_A\)

\(\Rightarrow\) max \(\text{dist}(i_A, v) \leq 3M - 1\)

In particular, dist \((i_A, i_D) \leq 3M - 1\)

Consider shortest path $\Pi$ from $i_A$ to $i_D$.

First edge cannot be of the form \((i_A, v)\) with weight $3M - 1$ (by rule (2)).

Because $v \neq i_D$ by construction of $H$ and any other edge

\((v, v')\) will have weight $\geq M$ making $\Pi$ have weight $\geq 3M - 1$.

Thus only path from $i_A$ to $i_D$ is $\Pi$ must be of the form $i_A \to j_B \to k_C \to i_D$ for some $j$ and $k \Rightarrow$ neg. triangle $i_A, k$,