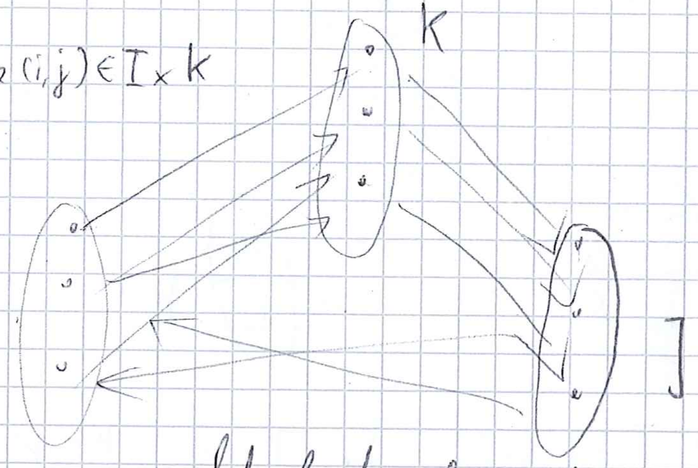
From ~~3-Layer APSP~~ Min-Plus Product to All-Pairs Negative Triangle

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

edge $(i,j) \in I \times K$

if



weight of edges from J to I is adaptive

<u>Goal</u>: $\forall i,j$ compute $\min_k (w(i,k), w(k,j))$

= minimum integer z s.t. there is a k with $w(i,k) + w(k,j) < z+1$

<u>Idea</u>: ~~Find z by~~ For every i,j, we can test whether

$\exists k: \; w(i,k) + w(k,j) < z(i,j) + 1$

$\Leftrightarrow \exists k: \; w(i,k) + w(k,j) + (-z(i,j) - 1) < 0$

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

$\rightarrow$ Find min. such $z(i,j)$ by simultaneous binary search

Weights are in $\{-n^c, \dots, n^c\}$

$\rightarrow$ each entry of Min-Plus Product is in $\{-2n^c, \dots, 2n^c\}$

$\Rightarrow$ binary search takes $\log_2(4n^c + 1) = O(\log n)$ steps

$\Rightarrow T(n)$-time alg. for All-Pairs Neg $\Delta \Rightarrow T(n) \log n$-time for Min-Plus

<u>Algorithm</u>: $\forall i,j$ initialize $m[i,j] = -2n^c$ and $M[i,j] = 2n^c$

repeat $\log(4n^c)$ times:

$\forall i,j: \; w(j,i) := -\lceil m(i,j) + M(i,j)/2 \rceil$

Compute All-Pairs Neg. Triangle

$\forall i,j$ if i,j is in neg. triangle: $M(i,j) := -w(j,i) - 1$

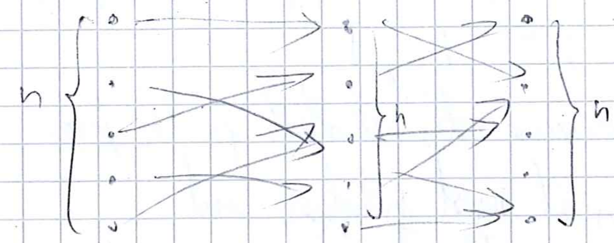
otherwise: $m(i,j) := -w(j,i)$

$\forall i,j$ set $C_{ij} = M(i,j)$

From APSP to Min-Plus Product $C = A \otimes B$

The reduction Min-Plus Product $\to$ APSP would be 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

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

Theorem: Let A be the weighted adjacency matrix of a graph G with nodes $V = \{v_1, \dots, v_n\}$ and let B be the same as A with every diagonal entry set to 0, i.e. $B = \underset{\uparrow \text{ pairwise min.}}{A \oplus} \underset{\begin{pmatrix} 0 & & \infty \\ & \ddots & \\ \infty & & 0 \end{pmatrix}}{I}$ in the min-plus semiring.

Then in the matrix $C = B^{2^{\lceil \log n \rceil}}$ (with exponentiation according to min-plus product) contains, for every i, j, entry $C[i,j]$ = length of shortest path from v_i to v_j.

Proof: We prove by induction on h: For all i, j

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

Base Case $h=0$: $B^0 = I$ ✓ (every node is at distance 0 from itself)

Inductive Step: $h \to h+1$: $B^{h+1} = B^h \otimes B = B^h \otimes (A \oplus I)$

$$= (B^h \otimes A) \oplus (B^h \otimes I) = (B^h \otimes A) \oplus B^h$$

By IH: B^h contains lengths of shortest paths with $\le h$ edges

Every shortest path with $\le h+1$ edges either consists of $\le h$ edges, ("B^h")

or first a shortest path with $\le h$ edges and then a single edge ("$B^h \otimes A$")

~~Thus B^{h+1} is the desired matrix~~

~~The Theorem now follows because~~

Thus: $dist^{\leq h+1}(v_i, v_j) = \min\left(\min_k\left(dist^h(v_i, v_k) + w(v_k, v_j)\right), dist^h(v_i, v_k)\right)$

$= \min\left((B^h \otimes A)[i,j], B^h[i,j]\right)$

$= B^{h+1}[i,j]$

The Theorem now follows because the shortest path from v_i to v_j (for all i, j) has at most $n-1$ edges and

$n-1 \leq 2^{\lceil \log n \rceil}$ $\square$

How to compute $B^{2^{\lceil \log n \rceil}}$?

Repeated squaring:

B

$B^2 = B \otimes B$

$B^4 = B^2 \otimes B^2$

$\vdots$

$B^{2^{\lceil \log n \rceil}} = B^{2^{\lceil \log n \rceil - 1}} \otimes B^{2^{\lceil \log n \rceil - 1}}$

$\leq \log\left(2^{\lceil \log n \rceil}\right) = O(\log n)$ many min-plus products

<u>Thm</u>: If Min-Plus Product has a $T(n)$-time algorithm, then APSP has an $O((T(n) + n^2)\log n)$-time algorithm

<u>Remark</u>: The $\log n$ factor can be eliminated (see [Aho / Hopcroft / Ullman '74])

<u>Remark</u>: In the APSP problem, non-edges, i.e. pairs of nodes u, v s.t. $(u,v) \notin E$, can be simulated by

(a) adding an edge (u, v) of weight ∞ 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 $\leq (n-1) W$

If shortest path after adding edges in step (b) has length $> (n-1) W$, we know that initially no path existed

Subcubic Equivalence of Radius

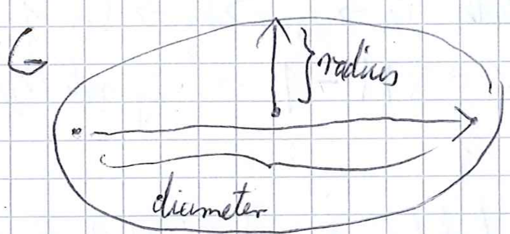
Definition (Radius):

Input: Weighted directed graph G with ~~x~~ edge weights $\in \{0,1,\dots,n^c\}$ for some constant c

Task: Output $\min_u \max_v$ ~~dist(u,v)~~ $\mathrm{dist}(u,v)$

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

Intuition: u is the ~~max~~ most central vertex



$\max_v \mathrm{dist}(u,v)$ is also called eccentricity of u

We will ~~x~~ prove:

Thm: APSP $\equiv$ Radius (subcubic equivalent)

by showing

Neg. Triangle $\leq$ Radius and

Radius $\leq$ APSP

Reduction: Radius $\rightarrow$ APSP

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

$\rightarrow$ subcubic reduction with one oracle call

Reduction: Negative Triangle $\rightarrow$ Radius

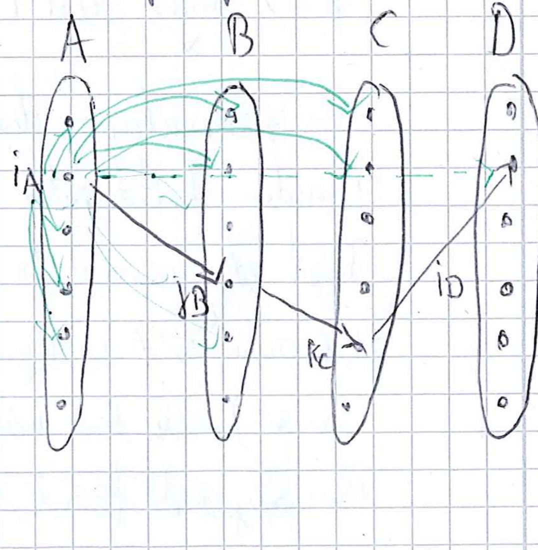
Given: directed graph with n nodes $\{1,\dots,n\}$ and edge weights in $\{-M,-M+1,\dots,M\}$ where $M = n^c$ for some constant c

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

(1) Make 4 layers A,B,C,D with n nodes each $A = \{1_A,\dots,n_A\}$, etc.

(2) For every edge (i,j) of G: Add (i_A,j_B), (j_B,j_C), (i_C,j_D) of weight $M+w(i,j)$ to H

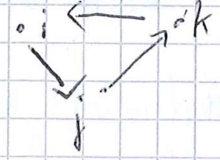
(3) For every $1 \leq i \leq n$, add edges of weight $3M-1$ from i_A to all other nodes, except i_D and i_A



Lemma: ~~The radius of H is $\le 3M-1$ iff~~

G contains a negative triangle iff the radius of H is $\le 3M-1$

Proof:

"$\Rightarrow$" Consider triangle  of total weight $W \le -1$

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

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

For any other vertex $v \neq i_D$: $dist(i_A, v) \le 3M-1$ by (3)

$\Rightarrow \max_v dist(i_A, v) \le 3M-1$

$\Rightarrow \min_u \max_v dist(u,v) \le 3M-1$ ✓

"$\Leftarrow$" We have $\min_u \max_v d(u,v) \le 3M-1$

If $u \neq i_A$ for some $1 \le i \le n$, then

$dist(u, i'_A) = \infty$ for any i'

$\Rightarrow \max_v dist(u,v) = \infty$

Thus $\min_u \max_v dist(u,v)$ is minimized by some $u = i_A$

$\Rightarrow \max_v dist(i_A, v) \le 3M-1$

in particular: $dist(i_A, i_D) \le 3M-1$

Consider shortest path π from i_A to i_D

First edge cannot be of the form (i_A, v) with weight $3M-1$ (i.e. rule (3) edge) because the $v \neq i_D$ by construction of H and any other edge (v, v') will have weight $\ge M$ making π have weight $> 3M-1$

Thus ~~only path from i_A to i_D~~ π 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, j, k