Complexity Theory of Polynomial-Time Problems

Lecture 5: Subcubic Equivalences

Karl Bringmann
Reminder: Relations = Reductions

transfer hardness of one problem to another one by reductions

\[ I \text{ is a 'yes'-instance} \quad \iff \quad J \text{ is a 'yes'-instance} \]

\[ t(n) \text{ algorithm for } Q \text{ implies a } r(n) + t(s(n)) \text{ algorithm for } P \]

if \( P \) has no \( r(n) + t(s(n)) \) algorithm then \( Q \) has no \( t(n) \) algorithm
Reminder: Relations = Reductions

Problem: $P$
Instance: $I$
Size: $n$

Reduction: $r(n)$
Total time:

Problem: $Q$
Instance: $I_1$, $I_2$, ..., $I_k$
Size: $n_1$, $n_k$
A subcubic reduction from P to Q is an algorithm A for P with oracle access to Q s.t.:

- For any instance I, algorithm A(I) correctly solves problem P on I.
- A runs in time $r(n) = O(n^{3-\gamma})$ for some $\gamma > 0$.
- For any $\varepsilon > 0$ there is a $\delta > 0$ s.t. $\sum_{i=1}^{k} n_i^{3-\varepsilon} \leq n^{3-\delta}$.
A subcubic reduction from P to Q is an algorithm A for P with oracle access to Q with:

A subcubic reduction implies:

If Q has an $O(n^{3-\alpha})$ algorithm for some $\alpha > 0$,
then P has an $O(n^{3-\beta})$ algorithm for some $\beta > 0$

Properties:

for any instance I, algorithm $A(I)$ correctly solves problem P on I
A runs in time $r(n) = O(n^{3-\gamma})$ for some $\gamma > 0$
for any $\varepsilon > 0$ there is a $\delta > 0$ s.t. $\sum_{i=1}^{k} n_i^{3-\varepsilon} \leq n^{3-\delta}$

similar: subquadratic/subquartic reductions
Subcubic Reduction

**subcubic reduction:** write \( P \leq Q \)

**subcubic equivalent:** write \( P \equiv Q \) if \( P \leq Q \) and \( Q \leq P \)

**Transitivity:** (Exercise)

For problems \( A, B, C \) with \( A \leq B \) and \( B \leq C \) we have \( A \leq C \).

In particular: If \( A \leq B \) and \( B \leq C \) and \( C \leq A \)
then \( A, B, C \) are subcubic equivalent.

**Lemma:** (without proof)

If \( A \leq B \) and \( B \) is in time \( O \left( \frac{n^3}{2^{\Omega(\log n)^{1/2}}} \right) \)
then \( A \) is in time \( O \left( \frac{n^3}{2^{\Omega(\log n)^{1/2}}} \right) \).
Reminder

All-Pairs-Shortest-Paths (APSP):

given a weighted directed graph $G$, compute the (length of the) shortest path between any pair of vertices

each edge has a weight in $\{1, \ldots, n^c\}$

Floyd-Warshall‘62: $O(n^3)$

..., Williams‘14: $O\left(n^3 / 2^\Omega(\log n)^{1/2}\right)$

Conjecture: for any $\epsilon > 0$ APSP has no $O(n^{3-\epsilon})$ algorithm

there exists $c > 0$ such that
Min-Plus Matrix Product:

Given $n_1 \times n_2$-matrix $A$ and $n_2 \times n_3$-matrix $B$, define their min-plus product as the $n_1 \times n_3$-matrix $C$ with

$$C_{i,j} = \min_{1 \leq k \leq n_2} A_{i,k} + B_{k,j}$$

From definition: $O(n^3)$ (if $n = n_1 = n_2 = n_3$)

Conjecture: for any $\varepsilon > 0$ there is no $O(n^{3-\varepsilon})$ algorithm.

There exists $c > 0$ such that
Reminder

Thm:
If APSP has a $T(n)$ algorithm then Min-Plus Product has an $O(T(n) + n^2)$ algorithm.

Thm:
If Min-Plus Product has a $T(n)$ algorithm then APSP has an $O((T(n) + n^2) \log n)$ algorithm.

Consider adjacency matrix $A$ of $G$

Add selfloops with cost 0: $A + I$

Square $\lceil \log n \rceil$ times using Min-Plus Product:

$B := (A + I)^{\lceil \log n \rceil}$

Then $B_{i,j}$ is the length of the shortest path from $i$ to $j$
Subcubic Reduction

A subcubic reduction from P to Q is an algorithm $A$ for $P$ with oracle access to $Q$ with:

Properties:
- for any instance $I$, algorithm $A(I)$ correctly solves problem $P$ on $I$
- $A$ runs in time $r(n) = O(n^{3-\gamma})$ for some $\gamma > 0$
- for any $\epsilon > 0$ there is a $\delta > 0$ s.t. $\sum_{i=1}^{k} n_i^{3-\epsilon} \leq n^{3-\delta}$
Subcubic Equivalences

**Thm:** If APSP has a $T(n)$ algorithm then Min-Plus Product has an $O(T(n))$ algorithm.

**Thm:** If Min-Plus Product has a $T(n)$ algorithm then APSP has an $O(T(n) \log n)$ algorithm.

APSP and Min-Plus Product are subcubic equivalent

**Cor:** APSP has an $O(n^{3-\epsilon})$ algorithm for some $\epsilon > 0$ if and only if Min-Plus Product has an $O(n^{3-\delta})$ algorithm for some $\delta > 0$

**Cor:** Min-Plus Product is in time $O \left( \frac{n^3}{2^{\Omega(\log n)^{1/2}}} \right)$
Subcubic Equivalences

- APSP
- Min-Plus Product
- All-Pairs-Negative-Triangle
- Negative Triangle
- Betweenness Centrality
- Radius
- Median
- Maximum Submatrix
- Metricity
- Minimum Shortest Path

[Vassilevska-Williams, Williams’10]
[Abboud, Grandoni, Vassilevska-Williams’15]
Triangle Problems

Negative Triangle

Given a weighted directed graph $G$

Decide whether there are vertices $i, j, k$ such that

$$w(j, i) + w(i, k) + w(k, j) < 0$$

from definition: $O(n^3)$

no $O(n^{3-\varepsilon})$ algorithm known (which works for all $c > 0$)

Intermediate problem:

All-Pairs-Negative-Triangle

Given a weighted directed graph $G$ with vertex set $V = I \cup J \cup K$

Decide for every $i \in I, j \in J$ whether there is a vertex $k \in K$ s.t.

$$w(j, i) + w(i, k) + w(k, j) < 0$$
Subcubic Equivalences

2\textsuperscript{nd} Shortest Path

Maximum Submatrix

Metricity

APSP

Min-Plus Product

All-Pairs-Negative-Triangle

Negative Triangle

Betweenness Centrality

Radius

Median

\[ \text{[Vassilevska-Williams, Williams'10]} \]
\[ \text{[Abboud, Grandoni, Vassilevska-Williams'15]} \]
Neg-Triangle to Min-Plus-Product

Given a weighted directed graph $G$ on vertex set $\{1, \ldots, n\}$

Adjacency matrix $A$:

$A_{i,j} = \text{weight of edge } (i, j),$ or $\infty$ if the edge does not exist

1. Compute Min-Plus Product $B := A \ast A$:

$B_{i,j} = \min_k A_{i,k} + A_{k,j}$

2. Compute $\min_{i,j} A_{j,i} + B_{i,j}$

this equals $\min_{i,j,k} A_{j,i} + A_{i,k} + A_{k,j}$

i.e. the smallest weight of any triangle

thus we solved Negative Triangle

Running Time: $T_{\text{NegTriangle}}(n) \leq T_{\text{MinPlus}}(n) + O(n^2)$

$\rightarrow$ subcubic reduction
Subcubic Equivalences

- **2nd Shortest Path**
- **Maximum Submatrix**
- **Metricity**
- **APSP**
- **Min-Plus Product**
- **All-Pairs-Negative-Triangle**
- **Negative Triangle**
- **Betweenness Centrality**
- **Radius**
- **Median**

\[ Vassilevska-Williams, Williams'10 \]
\[ Abboud, Grandoni, Vassilevska-Williams'15 \]
Add all edges from J to I with (carefully chosen) weights \( w(j, i) \)

Run All-Pairs-Negative-Triangle algorithm

Result: for all \( i, j \), is there a \( k \) such that \( w(j, i) + w(i, k) + w(k, j) < 0 \)?

\[ \iff w(i, k) + w(k, j) < -w(j, i) \]

**WANTED:** Min-Plus: for all \( i, j \): \( \min_k w(i, k) + w(k, j) \)

= minimum number \( z \) s.t. there is a \( k \) s.t. \( w(i, k) + w(k, j) < z + 1 \)

**binary search** via \( w(j, i) \)! simultaneous for all \( i, j \)!
Min-Plus to All-Pairs-Neg-Triangle

\[
\begin{array}{ccccccc}
3 & 1 & \infty & \infty \\
\infty & \infty & \infty & 4 & \infty \\
\infty & \infty & \infty & \infty & 2 \\
\infty & \infty & \infty & \infty & \infty & 1 \\
\end{array}
\]

\[
\begin{array}{ccccccc}
5 & \infty & \infty & \infty & \infty \\
7 & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & 2 \\
\infty & \infty & \infty & \infty & 4 \\
\end{array}
\]

\[
\begin{array}{ccccccc}
A & I & 2 & -2 & \infty & \infty \\
& & 1 & & -7 & \\
& & & & & & \\
& & & & & & \\
J & B & & & & & \\
\end{array}
\]

**binary search** via \( w(j, i) \)! **simultaneous** for all \( i, j \)!

need that all (finite) weights are in \( \{-n^c, ..., n^c\} \)

each entry of Min-Plus Product is in \( \{-2n^c, ..., 2n^c, \infty\} \)

binary search takes \( \log_2(4n^c + 1) = O(\log n) \) steps
**Min-Plus to All-Pairs-Neg-Triangle**

**Diagram:**
- **A** and **B** are matrices with values.
- **K** is a node in the diagram.
- **I** and **J** are paths connecting nodes.
- **n = 4** in the picture.

**Binary Search via** $w(j,i)$! **Simultaneous** for all $i,j$!

**Algorithm:**
- For all $i,j$: Initialize $m(i,j) := -2n^c$ and $M(i,j) := 2n^c$.
- Repeat $\log(4n^c)$ times:
  - For all $i,j$: Set $w(j,i) := -[(m(i,j) + M(i,j))/2]$.
  - Compute All-Pairs-Negative-Triangle.
- For all $i,j$: If $i,j$ is in negative triangle: $M(i,j) := -w(j,i) - 1$.
  Otherwise: $m(i,j) := -w(j,i)$.

**Notes:**
- Handling of $\infty$.

**Min-Plus Product**

**All-Pairs-Negative-Triangle**
Min-Plus to All-Pairs-Neg-Triangle

$$\begin{array}{cccccc}
3 & 1 & \infty & \infty & \\
\infty & \infty & 4 & \infty & \\
\infty & \infty & \infty & 2 & \\
\infty & \infty & \infty & \infty & 1 \\
\end{array}$$

$$\begin{array}{cccccc}
5 & \infty & \infty & \infty & \\
7 & \infty & \infty & \infty & \\
\infty & 2 & \infty & \infty & \\
\infty & \infty & \infty & 4 & \\
\end{array}$$

$$A$$

$$K$$

$$B$$

$$\begin{array}{cccccc}
3 & 1 & \\
5 & 7 & \\
4 & 2 & \\
\infty & \infty & \\
\infty & \infty & \\
\infty & \infty & \\
\end{array}$$

$$n = 4$$ in the picture

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

$$T(n)$$ algorithm for All-Pairs-Neg-Triangle yields $$O(T(n) \log n)$$ algorithm for Min-Plus Product

In particular: $$O(n^{3-\varepsilon})$$ algorithm for All-Pairs-Neg-Triangle for some $$\varepsilon > 0$$ implies $$O(n^{3-\varepsilon})$$ algorithm for Min-Plus Product for some $$\varepsilon > 0$$

$$\rightarrow$$ subcubic reduction
Subcubic Equivalences

2\textsuperscript{nd} Shortest Path

Maximum Submatrix

Metricity

APSP

Min-Plus Product

All-Pairs-Negative-Triangle

Negative Triangle

Betweenness Centrality

Radius

Median

[Vassilevska-Williams, Williams’10]

[Abboud, Grandoni, Vassilevska-Williams’15]
All-Pairs-Neg-Triangle to Neg-Triangle

**Negative Triangle** Given graph $G$
Decide whether there are vertices $i, j, k$ such that
\[ w(j, i) + w(i, k) + w(k, j) < 0 \]

**All-Pairs-Negative-Triangle** Given graph $G$ with vertex set $V = I \cup J \cup K$
Decide for every $i \in I, j \in J$ whether there is a vertex $k \in K$ such that
\[ w(j, i) + w(i, k) + w(k, j) < 0 \]

Split $I, J, K$ into $n/s$ parts of size $s$:
\[ I_1, ..., I_{n/s}, J_1, ..., J_{n/s}, K_1, ..., K_{n/s} \]

For each of the $(n/s)^3$ triples $(I_x, J_y, K_z)$:
consider graph $G[I_x \cup J_y \cup K_z]$
All-Pairs-Neg-Triangle to Neg-Triangle

Initialize $C$ as $n \times n$ all-zeroes matrix

For each of the $(n/s)^3$ triples of parts $(I_x, J_y, K_z)$:

While $G[I_x \cup J_y \cup K_z]$ contains a negative triangle:

Find a negative triangle $(i, j, k)$ in $G[I_x \cup J_y \cup K_z]$

Set $C[i, j] := 1$

Set $w(i, j) := \infty$

$(i, j)$ is in no more negative triangles

✓ guaranteed termination:
  can set $\leq n^2$ weights to $\infty$

✓ correctness:
  if $(i, j)$ is in negative triangle, we will find one
All-Pairs-Neg-Triangle to Neg-Triangle

Find a negative triangle \((i, j, k)\) in \(G[I_x \cup J_y \cup K_z]\)

How to **find** a negative triangle
*if we can only decide whether one exists?*

Partition \(I_x\) into \(I_x^{(1)}, I_x^{(2)}\), \(J_y\) into \(J_y^{(1)}, J_y^{(2)}\), \(K_z\) into \(K_z^{(1)}, K_z^{(2)}\)

Since \(G[I_x \cup J_y \cup K_z]\) contains a negative triangle, at least one of the \(2^3\) subgraphs
\(G[I_x^{(a)} \cup J_y^{(b)} \cup K_z^{(c)}]\)
contains a negative triangle

Decide for each such subgraph whether it contains a negative triangle

Recursively find a triangle in one subgraph
All-Pairs-Neg-Triangle to Neg-Triangle

Find a negative triangle \((i,j,k)\) in \(G[I_x \cup J_y \cup K_z]\)

How to find a negative triangle if we can only decide whether one exists?

Partition \(I_x\) into \(I_x^{(1)}, I_x^{(2)}\), \(J_y\) into \(J_y^{(1)}, J_y^{(2)}\), \(K_z\) into \(K_z^{(1)}, K_z^{(2)}\)

Since \(G[I_x \cup J_y \cup K_z]\) contains a negative triangle, at least one of the \(2^3\) subgraphs

\[ G[I_x^{(a)} \cup J_y^{(b)} \cup K_z^{(c)}] \]

contains a negative triangle

Decide for each such subgraph whether it contains a negative triangle

Recursively find a triangle in one subgraph

Running Time:

\[ T_{\text{FindNegTriangle}}(n) \leq 2^3 \cdot T_{\text{DecideNegTriangle}}(n) + T_{\text{FindNegTriangle}}(n/2) \]

\[ = O(T_{\text{DecideNegTriangle}}(n)) \]
All-Pairs-Neg-Triangle to Neg-Triangle

Initialize \( C \) as \( n \times n \) all-zeroes matrix

For each of the \((n/s)^3\) triples of parts \((I_x,J_y,K_z)\):

While \( G[I_x \cup J_y \cup K_z] \) contains a negative triangle:

Find a negative triangle \((i,j,k)\) in \( G[I_x \cup J_y \cup K_z] \)

Set \( C[i,j] := 1 \)

Set \( w(i,j) := \infty \)

Running Time:

\[
(\ast) = O(T_{\text{FindNegTriangle}}(s)) = O(T_{\text{DecideNegTriangle}}(s))
\]

Total time: \((\#\text{triples}) + (\#\text{triangles found}) \cdot (\ast)\)

\[\leq ((n/s)^3 + n^2) \cdot T_{\text{DecideNegTriangle}}(s)\]

Set \( s = n^{1/3} \) and assume \( T_{\text{DecideNegTriangle}}(n) = O(n^{3-\varepsilon})\)

Total time: \( O(n^2 \cdot n^{1-\varepsilon/3}) = O(n^{3-\varepsilon/3})\)
Subcubic Equivalences

\[ \text{Negative Triangle} \iff \text{All-Pairs-Negative-Triangle} \]

\[ \text{Min-Plus Product} \iff \text{APSP} \]

- 2\textsuperscript{nd} Shortest Path
- Maximum Submatrix
- Metricity
- Betweenness Centrality
- Radius
- Median

\[ \text{[Vassilevska-Williams, Williams'10]} \]
\[ \text{[Abboud, Grandoni, Vassilevska-Williams'15]} \]
Radius

\( G \) is a weighted directed graph
\( d(u, v) \) is the distance from \( u \) to \( v \) in \( G \)

**Radius:** \( \min_u \max_v d(u, v) \)

\( u \) is in some sense the *most central vertex*

compute all pairwise distances,
then evaluate definition of radius in time \( O(n^2) \)

→ subcubic reduction

\( \Rightarrow \) Radius is in time \( O \left( n^3 / 2^{\Omega(\log n)^{1/2}} \right) \)
Negative Triangle to Radius

Negative Triangle instance:
graph $G$ with $n$ nodes,
edge-weights in $\{-n^c, ..., n^c\}$

Radius instance:
graph $H$ with $O(n)$ nodes,
edge-weights in $\{0, ..., O(n^c)\}$

1) Make four layers with $n$ nodes
2) For any edge $(i, j)$: Add $(i_A, j_B)$,
   $(i_B, j_C), (i_C, j_D)$ with weight $M + w(i, j)$
Negative Triangle to Radius

Negative Triangle instance:
graph $G$ with $n$ nodes, edge-weights in $\{-n^c, ..., n^c\}$

Radius instance:
graph $H$ with $O(n)$ nodes, edge-weights in $\{0, ..., O(n^c)\}$

$(i, j, k)$ has weight $W$

1) Make four layers with $n$ nodes
2) For any edge $(i, j)$: Add $(i_A, j_B)$, $(i_B, j_C), (i_C, j_D)$ with weight $M + w(i, j)$

$\Leftrightarrow$ path has length $3M + W$

$\rightarrow$ exists $i_A, j_B, k_C, i_D$-path of length $\leq 3M - 1$
**Negative Triangle to Radius**

**Negative Triangle instance:**
graph $G$ with $n$ nodes, edge-weights in $\{-n^c, \ldots, n^c\}$

$$(i,j,k) \text{ has weight } W$$

1) Make four layers with $n$ nodes
2) For any edge $(i,j)$: Add $(i_A,j_B)$, $(i_B,j_C),(i_C,j_D)$ with weight $M + w(i,j)$
3) Add edges of weight $3M - 1$ from any $i_A$ to all nodes except $i_D$ (and $i_A$)

**Radius instance:**
graph $H$ with $O(n)$ nodes, edge-weights in $\{0, \ldots, O(n^c)\}$

$M := 3n^c$

$$\Leftrightarrow \text{path has length } 3M + W$$

$$\rightarrow \exists i_A,j_B,k_C,i_D-\text{path of length } \leq 3M - 1?$$

**Claim:** Radius of $H$ is $\leq 3M - 1$ iff there is a negative triangle in $G$
**Negative Triangle to Radius**

**Claim:** Radius of $H$ is $\leq 3M - 1$ iff there is a negative triangle in $G$

**Proof:**

If there is a negative triangle $(i, j, k)$ then $i_A$ is in distance $\leq 3M - 1$ to $i_D$ (by (2)), and in distance $\leq 3M - 1$ to any other vertex (by (3)), so the radius is $\leq \max_v d(i_A, v) \leq 3M - 1$

If there is no negative triangle $(i, j, k)$:

Any node $u$ of the form $i_B/i_C/i_D$ cannot reach $A$, so it has $\max_v d(u, v) = \infty$

Any $i_A$ is in distance $\geq 3M$ to $i_D$, since there is no $i_A, j_B, k_C, i_D$-path of length $\leq 3M - 1$ (note that the edges added in (3) also do not help)

Hence, for all $u$, $\max_v d(u, v) \geq 3M$, and thus the radius is at least $3M$
Subcubic Equivalences

2\textsuperscript{nd} Shortest Path

Maximum Submatrix

Metricity

APSP \iff Min-Plus Product \iff All-Pairs-Negative-Triangle \iff Negative Triangle

Betweenness Centrality

Radius

Median

[Vassilevska-Williams, Williams’10]
[Abboud, Grandoni, Vassilevska-Williams’15]
MaxSubmatrix

MaxSubmatrix:
given an $n \times n$ matrix $A$ with entries in $\{-n^c, \ldots, n^c\}$
$\Sigma(B) := \text{sum of all entries}$ of matrix $B$
compute maximum $\Sigma(B)$ over all submatrices $B$ of $A$

Thm: MaxSubmatrix is subcubic equivalent to APSP

there are $O(n^4)$ possible submatrices $B$
computing $\Sigma(B)$: $O(n^2)$
trivial running time: $O(n^6)$

Exercise: design an $O(n^3)$ algorithm

[Tamaki, Tokuyama’98]
[Backurs, Dikkala, Tzamos’16]
MaxSubmatrix

MaxSubmatrix:

given an \( n \times n \) matrix \( A \) with entries in \( \{-n^c, \ldots, n^c\} \)

\[ \Sigma(B) := \text{sum of all entries} \] of matrix \( B \)

compute maximum \( \Sigma(B) \) over all \textbf{submatrices} \( B \) of \( A \)

Thm: MaxSubmatrix is subcubic equivalent to APSP

MaxCenteredSubmatrix:

compute maximum \( \Sigma(B) \) over all \textbf{submatrices} \( B \) of \( A \) containing the center of \( A \)

i.e. we require \( x_1 \leq n/2 < x_2 \) and \( y_1 \leq n/2 < y_2 \)

Thm: MaxCenteredSubmatrix is subcubic equ. to APSP

we only prove: NegativeTriangle \( \leq \) MaxCenteredSubmatrix

Exercise: MaxCenteredSubmatrix \( \leq \) APSP
**NegTriangle** to **MaxCentSubmatrix**

**Positive** Triangle instance:
- graph $G$ with $n$ nodes,
- edge-weights in $\{-n^c, ..., n^c\}$

MaxCenteredSubmatrix:
- $2n \times 2n$-matrix $A$
- entries in $\{-n^{O(c)}, ..., n^{O(c)}\}$

In quadrant II we want for any $k, i$:
\[
\sum_{y=k}^{n} \sum_{x=i}^{n} A_{y,x} = w(k, i)
\]

this is satisfied by defining:
\[
A_{k,i} := w(k, i) - w(k + 1, i) - w(k, i + 1) + w(k + 1, i + 1)
\]

(\text{where } w(x, y) := 0 \text{ for } x > n \text{ or } y > n)
NegTriangle to MaxCentSubmatrix

**Positive** Triangle instance:
graph $G$ with $n$ nodes,
edge-weights in $\{-n^c, \ldots, n^c\}$

MaxCenteredSubmatrix:
$2n \times 2n$-matrix $A$
entries in $\{-n^O(c), \ldots, n^O(c)\}$

In quadrant II we want for any $k, i$:
$$\sum_{y=k}^{n} \sum_{x=i}^{n} A_{y,x} = w(k, i)$$

this is satisfied by defining:
$$A_{k,i} := w(k, i) - w(k + 1, i)$$
$$-w(k, i + 1) + w(k + 1, i + 1)$$

(where $w(x, y) := 0$ for $x > n$ or $y > n$)

With this definition of $A$, for any $1 \leq k, i \leq n$:
$$\sum_{y=k}^{n} \sum_{x=i}^{n} A_{y,x} = \sum_{y=k}^{n} \sum_{x=i}^{n} w(y, x) - w(y + 1, x) - w(y, x + 1) + w(y + 1, x + 1)$$

where any $w(y, x)$ with $k < y \leq n$ and $i < x \leq n$
appears with factors $+1 - 1 - 1 + 1 = 0$

and any $w(y, x)$, s.t. exactly one of $y = k$ or
$y = n + 1$ or $x = i$ or $x = n + 1$ holds,
appears with factors $+1 - 1 = 0$

and since $w(y, n + 1) = w(n + 1, x) = 0$,
the only remaining summand is $w(k, i)$
\textbf{Positive} Triangle instance: graph $G$ with $n$ nodes, edge-weights in $\{-n^c, \ldots, n^c\}$

MaxCenteredSubmatrix: $2n \times 2n$-matrix $A$ entries in $\{-n^O(c), \ldots, n^O(c)\}$

Claim: MaxCentSubmatrix of $A$ is $> M$ iff $G$ has a \textit{positive} triangle

In quadrant II we want for any $k, i$:

$$\sum_{y=k}^{n} \sum_{x=i}^{n} A_{y, x} = w(k, i)$$

this is satisfied by defining:

$$A_{k, i} := w(k, i) - w(k + 1, i)$$

$$-w(k, i + 1) + w(k + 1, i + 1)$$

(where $w(x, y) := 0$ for $x > n$ or $y > n$)
Summary

- APSP
- Min-Plus Product
- All-Pairs-Negative-Triangle
- Negative Triangle
- 2nd Shortest Path
- Maximum Submatrix
- Metricity
- Betweenness Centrality
- Radius
- Median

\[ \text{Metricity} \iff \text{Negative Triangle} \iff \text{All-Pairs-Negative-Triangle} \iff \text{Min-Plus Product} \iff \text{APSP} \iff \text{2nd Shortest Path} \]

\[ \text{Maximum Submatrix} \]

\[ \text{Betweenness Centrality} \]

\[ \text{Radius} \]

\[ \text{Median} \]