A Quick Intro to the APSP Hypothesis

Jakob Nogler[1]

[1] Massachusetts Institute of Technology

APSP stands for: All-Pairs Shortest Path Problem

- **Input:** A graph $\mathbf{G} = (V, E)$, possibly weighted and directed. Input size: $O(n + m)$
- **Goal:** Compute shortest path distance $d(u, v)$ for each pair $u, v \in V$. Output size: $O(n^2)$
- Trivial algorithm takes time $O(n \cdot (m + n \log n)) = O(nm + n^2 \log n)$.

Weights	Graph	Time
Unweighted	Undirected	$\tilde{O}(n^\omega)$ [Sei95]
Unweighted	Directed	$\tilde{O}(n^{2.53})$ [Zwi02]
Weighted	Undirected/Directed	$n^3 / 2^{\Omega(\sqrt{\log n})}$ [Wil14]
Weighted	Undirected/Directed	$O(mn + n^2 \log n)$

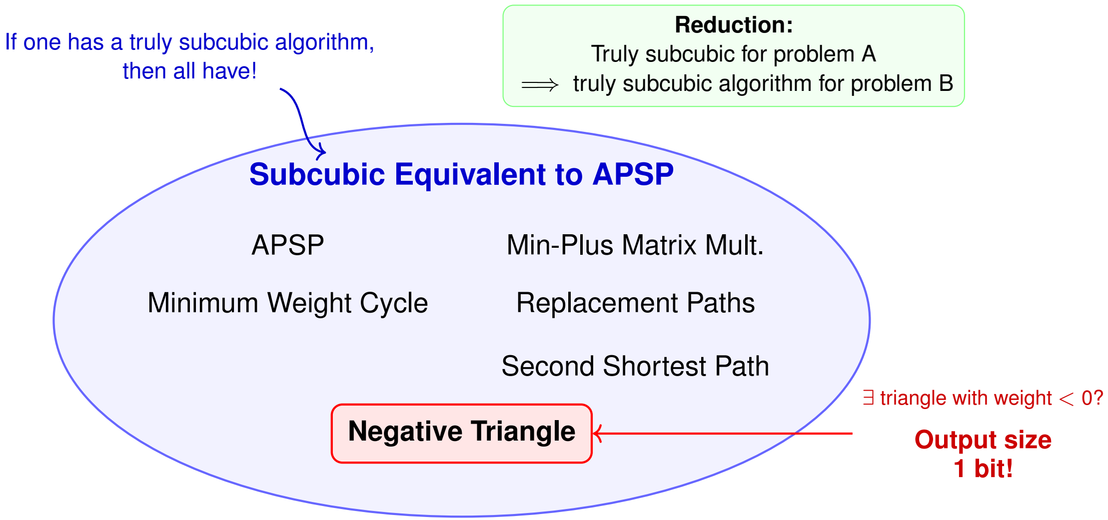
} Matrix multiplication exponent $\omega = 2.3713...$

} **Open Question:** Is there truly subcubic algorithm? e.g. $O(n^{2.9999})$.

Time	Author(s)	Year(s)
n^3	Floyd / Warshall	1962/1962
$n^3 / \log^{1/3} n$	Fredman	1975
$n^3 / \log^{1/2} n$	Dobosiewicz / Takaoka	1990/1991
$n^3 / \log^{5/7} n$	Han	2004
$n^3 / \log n$	Takaoka / Zwick / Takaoka / Chan	2004/2004/2005/2005
$n^3 / \log^{5/4} n$	Han	2006
$n^3 / \log^2 n$	Chan / Han-Takaoka	2007/2012
$n^3 / 2^{\Omega(\log n)^{1/2}}$	Williams	2014

Subcubic Equivalences Between Path, Matrix, and Triangle Problems

Virginia Vassilevska Williams and Ryan Williams (FOCS 2010)

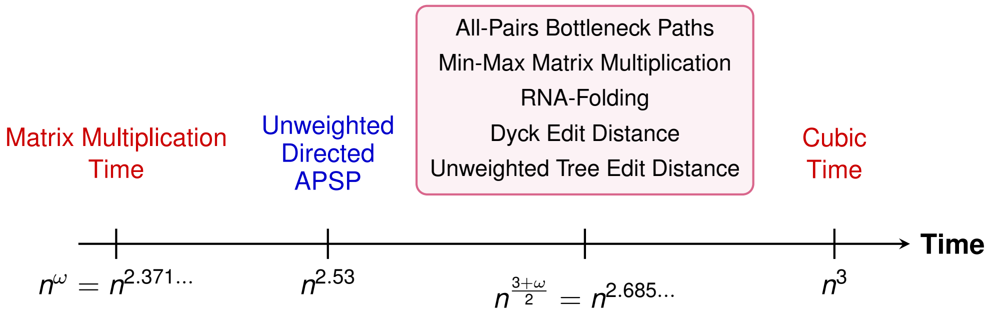


The APSP Conjecture

There is $c > 0$ s.t. for no $\varepsilon > 0$ APSP with weights in $\{-n^c, \ldots, n^c\}$ can be solved in $O(n^{3-\varepsilon})$ time in the Word RAM model.

- Computing the Radius of a Graph
- Computing the Median of a Graph
- Computing the Betweenness Centrality
- 2D maximum subarray problem
- Maximum-weight rectangle
- Tree edit distance
- Undirected Single Source Replacement Path
- Decremental Diameter
- Dynamic Planar APSP
- Distance Sensitivity Oracle
- Undirected Dual Fault Replacement Paths
- Weighted Edit Distance
- Zero Weight Triangle
- ...

Takeaway: If you cannot break the $O(n^3)$ time check for APSP-hardness!



Many Problems which are APSP hard have an unweighted version with complexity between n^ω and n^3

Approach for Many such Problems:

Prove APSP-hardness of weighted case $\Longrightarrow$ Focus on unweighted cases and find “right“ subcubic time.

My Takeaway

Don’t focus on unweighted case, focus on proving APSP equivalence instead!

Potential benefits:

- Yields a $n^3/2^{\Omega(\sqrt{\log n})}$ algorithm for free.
- Faster unweighted algorithms usually follow naturally.
- A “reduction” is a nicer result than a single algorithm.

(String) Edit Distance

Input: Strings S, S'. **Output:** Cheapest transformation of S into S'.

Substitution: `abcxef` → `abcyef`

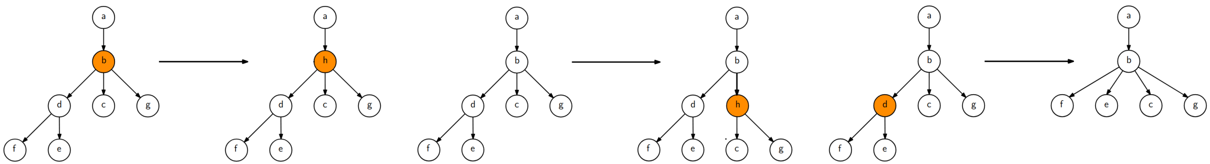
Insertion: `abcef` → `abcxef`

Deletion: `abcdef` → `abdef`

Tree Edit Distance (TED)

Input: Labeled, left-to-right ordered trees T, T'. **Output:** Cheapest transformation of T into T'.

Substitution Insertion Deletion



Year	Work	Setting	Complexity
1979	Tai	weighted	$\mathcal{O}(n^6)$
1989	Shasha, Zhang	weighted	$\mathcal{O}(n^4)$
1998	Klein	weighted	$\mathcal{O}(n^3 \log n)$
2007	Demaine, Mozes, Rossman, Weimann	weighted	$\mathcal{O}(n^3)$
2020	Bringmann, Gawrychowski, Mozes, Weinmann	weighted	APSP hard
2025	**N, Polak, Saha, Vassilevska Williams, Xu, Ye**	**weighted**	$n^3/2^{\Omega(\sqrt{\log n})}$
2022	Mao	unweighted	$\mathcal{O}(n^{2.9546})$
2023	Dürr	unweighted	$\mathcal{O}(n^{2.9148})$
2025	**N, Polak, Saha, Vassilevska Williams, Xu, Ye**	**unweighted**	$\mathcal{O}(n^{(\omega+3)/2}) = \mathcal{O}(n^{2.687})$

14:00–15:12
ICALP (Track A)

ICALP Session 7.1

14:00 Parallel Reachability and Shortest Paths on Non-Sparse Digraphs: Near-Linear Work and Sub-Square-Root Depth
Vikrant Ashvinkumar, Aaron Bernstein, Maximilian Probst Gutenberg, Thatchaphol Saranurak

14:24 Fast Shortest Paths in Graphs with Sparse Signed Tree Models and Applications
Édouard Bonnet, Colin Geniet, Eun Jung Kim, Sungmin Moon

14:48 Undirected Replacement Paths: Dual Fault Reduces to Single Source
Jakob Nogler, Virginia Vassilevska Williams

Open Problems: Distance Sensitivity Oracles (DSO)

- **Preprocessing Phase:** Graph $G = (V, E) \implies$ Data Structure
- **Query Phase** (s, t, e)**:**
  - Return in $\mathcal{O}(1)$ time: $d_{G \setminus e}(s, t)$ (Shortest $s \to t$ path avoiding failed edge e)
- **Goal:** Minimize Preprocessing Time
  - Lower bound: APSP time (since the oracle easily answers standard APSP queries)

Graph	**Weights**	**Best Preprocessing Time**
Directed	Unweighted	$\tilde{\mathcal{O}}(n^{2.52})$ [KS23]
Undirected	Unweighted	$\tilde{\mathcal{O}}(n^{2.52})$ [KS23]
Directed/Undirected	Weighted	$\tilde{\mathcal{O}}(n^{3})$ [BK09]