Fine-Grained Complexity
Lecture 1: Overview

October 25, 2021

Instructor: Amir Abboud

TA: Tomer Grossman

Website: http://www.weizmann.ac.il/math/AmirAbboud/course-fine-grained-complexity

HW every other week, final take-home exam.

Participation in class encouraged and expected.

Starting point: **NP-Hardness** is great!

NP-hard	in P
Protein Folding	Linear Programming
Travelling Salesman	All Pairs Shortest Paths
Subset Sum	Fourier Transform
k-SAT	CFG Parsing
	Edit-Distance
…	…

What about the problems inside P?

Is "polynomial = efficient" true?

Why care about $O(n), O(n^{1.5}), O(n^2), \ldots ?$

Here's one example where it matters...

Local Alignment

Input: two (DNA) sequences of length n and a scoring matrix.

AGCCCGTCTACGTGCAACCGGGGAAAGTATA
AAACGTGACGAGAGAGAGAACCCATTACGAA

Output: The optimal alignment of two substrings.

C C G – T C T A C G
C C C A T – T A C G
+1 +1 -0.5 -1 +1 -1 +1 +1 +1 +1 = +4.5

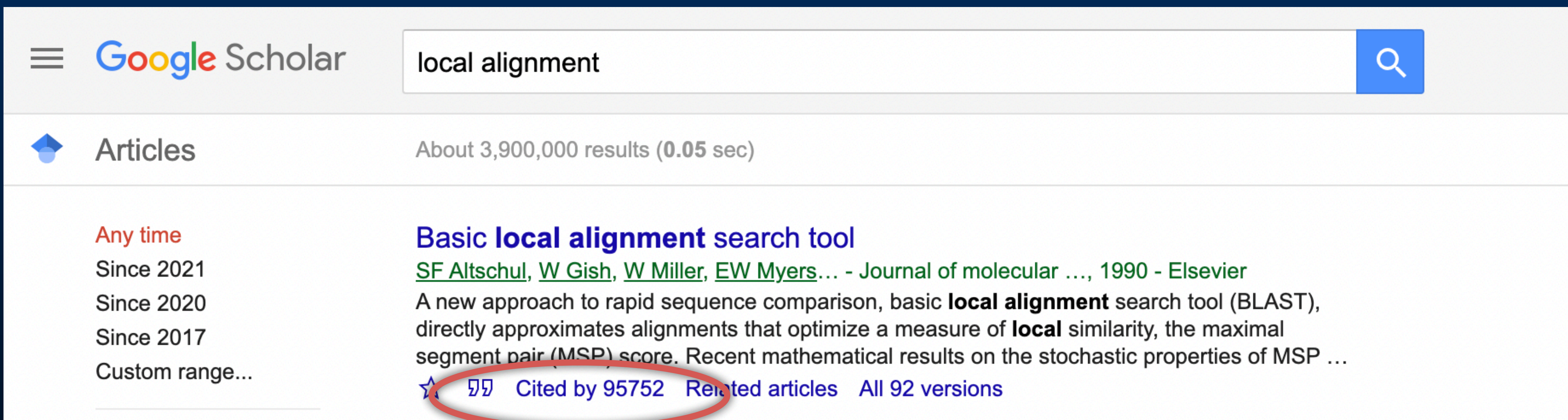
	A	C	G	T	–
A	+1	-1.4	-1.8	-0.7	-1
C	-1.4	+1	-0.5	-1	-1
G	-1.8	-0.5	+1	-1.9	-1
T	-0.7	-1	-1.9	+1	-1
-	-1	-1	-1	-1	$-\infty$

Typically: ***$n >> 10^6$***

[Smith-Waterman '81] *$O(n^2)$* with dynamic programming - too slow!

Why care about $O(n), O(n^{1.5}), O(n^2), \ldots?$

BLAST: A *heuristic, linear time* algorithm for **Local Alignment**.

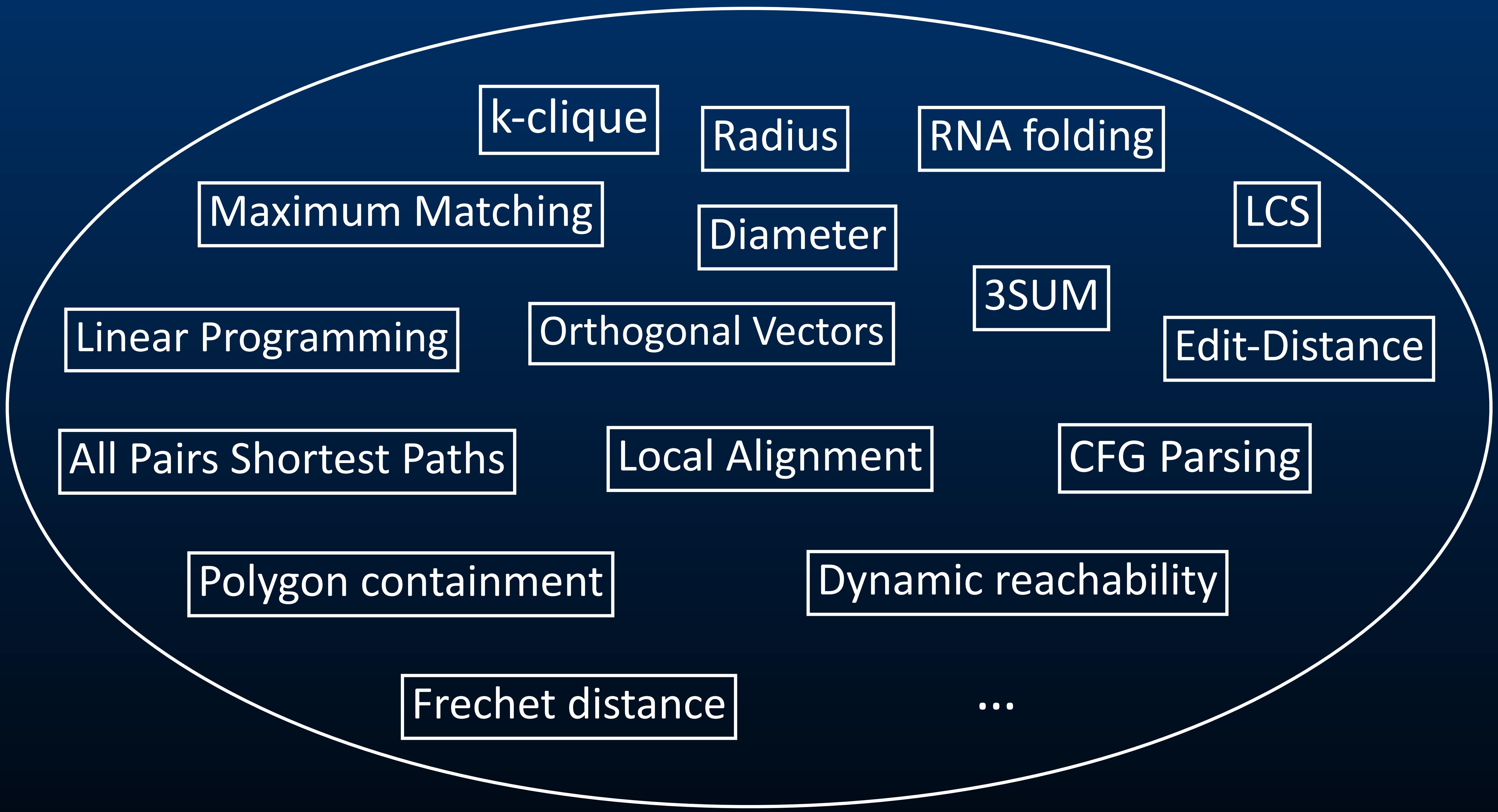


95k citations!

Are there fast algorithms with optimality guarantees?

Fine-Grained Complexity has the answers.

The Class P



Goal: *Understand the time complexity of important problems.*

Fine-Grained Complexity
or: Hardness in P

Take a problem X in P, say in ***$O(n^2)$*** time.

And prove that:

" X *probably* *cannot* be solved in $O(n^{2-\varepsilon})$ time. "

How do we get n^2 and n^3 lower bounds?

NP-hardness is not fine-grained enough…

Lower bounds for restricted algorithms?

e.g. $\Omega(n \log n)$ for sorting in the comparisons-only model.

Not general enough, and only gives partial answers.

Unconditional polynomial lower bounds?

"Any Turing Machine has to spend $\Omega(n^2)$ time…"

Time Hierarchy Thm (1965): Some (artificial) problems require $\Omega(n^2)$ time.

But $\Omega(n^2)$ for natural problems, even for SAT, is far out of reach of current techniques. Best lower bound is $3.1n$.

Fine-Grained Complexity

Take a problem X in P, say in ***O(n²)*** time.

And prove that:

" X *probably cannot* be solved in $O(n^{2-\varepsilon})$ time. "

Approach: imitate NP-hardness!

Theorem: Problem X is NP-hard.

X is in P $\Rightarrow$ Every NP-complete problem is in P *(unlikely)*

Conclusion: "X is probably not in P"

Approach: imitate NP-hardness!

To prove "lower bounds", reduce famous problems to your problem.

An $O(n^{1.9})$ algorithm for problem Y *(surprisingly fast)* $\Rightarrow$ Unexpected breakthroughs in different areas of CS

(some conjecture is refuted)

Conclusion: "Probably no $O(n^{1.9})$ algorithm for Y"

Next: Many examples of this approach in action.

An Example of a Fine-Grained Lower Bound

“No provably exact, fast algorithm.”

Theorem [AVW’14]:
“If for some $\varepsilon > 0$, we can solve Local Alignment in $O(n^{2-\varepsilon})$ time, then we can solve k-SAT in $O((2-\delta)^n)$ time for some $\delta > 0$ and all $k > 0$.”

Faster Local Alignment → Faster k-SAT → SETH is false

e.g. $O(n^{1.99})$ → e.g. $O(1.99^n)$

P ≠ NP: “k-SAT cannot be solved in polynomial time.”

The Strong Exponential Time Hypothesis (SETH):

“k-SAT cannot be solved even in $O(1.99^n)$ time.”

Today's Lecture

- Motivation
- Intro to Fine-Grained Complexity
- The Basics (Part 1: weeks 2 to 8)
- About this course
- Advanced topics (Part 2: weeks 8 to 14)

SETH

k-SAT: given a k-CNF formula on *n* variables and *m* clauses, is it satisfiable?

$$\phi = (x_1 \vee x_2 \vee \bar{x_3} \vee x_{10}) \wedge \cdots \wedge (x_2 \vee \bar{x_1} \vee x_4)$$

Fastest algorithms: [based on PPSZ'05]

$$O\left(2^{\left(1-\frac{1}{ck}\right)\cdot n}\right)$$

k=3: 1.308^n
k=4: 1.504^n
k=5: 1.592^n
... $k \to \infty : 2^n$

The Strong Exponential Time Hypothesis (SETH):
[Impagliazzo-Paturi'01]
There is no $\varepsilon > 0$ such that for all $k > 2$, **k-SAT** can be solved in $O((2-\varepsilon)^n)$ time.

SETH: "k-SAT cannot be solved in ***O(1.99^n)*** time."

Theorem [AVW'14]:

"If for some $\varepsilon > 0$, we can solve Local Alignment in $O(n^{2-\varepsilon})$ time, then we can solve k-SAT in $O((2-\delta)^n)$ time for some $\delta > 0$ and all $k > 0$."

Faster Local Alignment → Faster k-SAT → SETH is false

e.g. $O(N^{1.99})$

e.g. $O(1.99^n)$

P ≠ NP: "k-SAT cannot be solved in polynomial time."

The Strong Exponential Time Hypothesis (SETH):

"k-SAT cannot be solved even in $O(1.99^n)$ time."

Longest Common Subsequence (**LCS**)

Input: two sequences of length n

S = cddcabbabcbaa

T = adbdbbcabacdd

Output: the length of the longest common subsequence

Classic Dynamic Programming: $O(n^2)$

[Masek - Paterson ’80] $O(n^2 / \log^2 n)$

Longstanding open question:

Can we solve LCS in near-linear time?

Edit Distance

Input: two sequences of length n

S = cddcabbabcbaa → - - → → T = adbdbbcabacdd

Output: the min number of insertions/deletions/substitutions needed to transform one to the other

[Masek - Paterson ’80]

$O(n^2 / \log^2 n)$

Theorem [AVW'14]:

"If for some $\varepsilon > 0$, we can solve Local Alignment in $O(n^{2-\varepsilon})$ time, then we can solve k-SAT in $O((2-\delta)^n)$ time for some $\delta > 0$ and all $k > 0$."

Faster Local Alignment → Faster k-SAT → SETH is false

e.g. $O(N^{1.99})$

e.g. $O(1.99^n)$

[BI'15] Same for Edit Distance.

[ABV '15, BK'15] Same for LCS.

P ≠ NP: "k-SAT cannot be solved in polynomial time."

The Strong Exponential Time Hypothesis (SETH):

"k-SAT cannot be solved even in $O(1.99^n)$ time."

Some More SETH-based Lower Bounds

k-SAT

↓

Diameter
Closest Pair
Local Alignment
Dynamic Reachability
Single-Source Max-Flow
Subtree Isomorphism
Stable Matching
Edit-Distance
Frechet
LCS
…

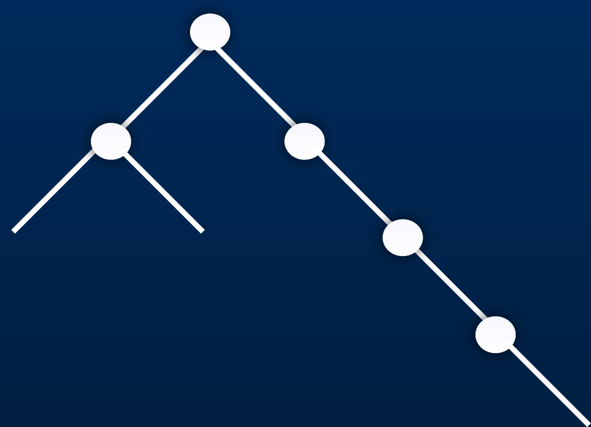
Problem domains:

Graph Algorithms,
Pattern Matching,
Computational Geometry,
High-dimensional Geometry,
Machine Learning,
Computational Biology,
Time-series analysis,
…

Subtree Isomorphism

(on binary trees)

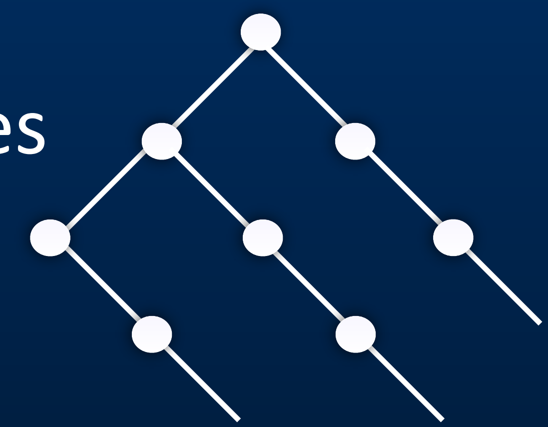
H



n nodes
depth h

G

O(n) nodes



"is H contained in G?"

Simple upper bound: $O(n^2)$

[ABHVZ'16]

Theorem: Subtree Isomorphism on binary trees in $O(n^{1.99})$ time refutes our SETH.

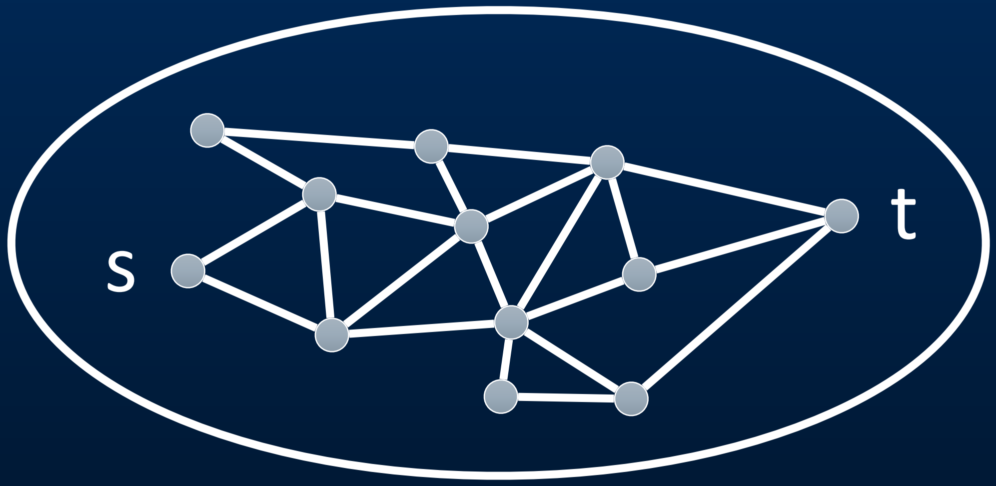
Dynamic Problems

Dynamic (undirected) Connectivity

Input: an undirected graph G

Updates: Add or remove edges.

Query: Are s and t connected?



Trivial algorithm: O(m) update/query time.

[Henzinger-King '95]:
$O(\log^2 n)$ amortized time per update.

Dynamic Problems

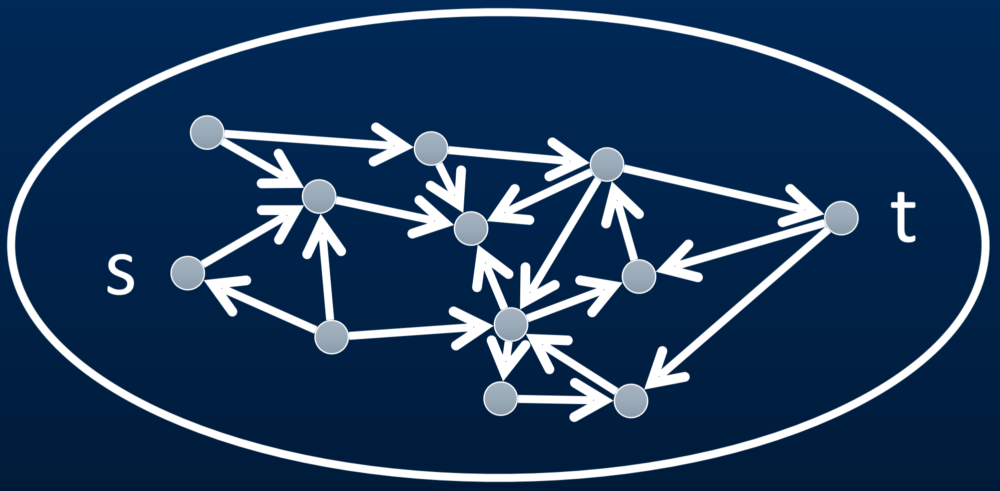
Dynamic (directed) Reachability

Input: A directed graph G.

Updates: Add or remove edges.

Query:

#SSR: How many nodes can s reach?



Trivial algorithm: O(m) time updates

[AV'14]

Theorem: If dynamic #SSR can be solved with **$O(m^{0.99})$** *amortized update* time then SETH is false.

Subclasses within P

k-SAT

- Diameter
- Closest Pair
- Local Alignment
- Dynamic Reachability
- Single-Source Max-Flow
- Subtree Isomorphism
- Stable Matching
- Edit-Distance
- Frechet
- LCS
- …

3SUM

- Colinearity
- Polygon Containment
- Strips Cover Rectangle
- Triangle Enumeration
- Compressed Inner Product
- Dynamic Max Matching
- Set Intersection
- …

APSP

- Radius
- Dynamic Max Matching
- Stochastic Context-Free Grammar Parsing
- Negative Triangle
- Dynamic Max Flow
- Replacement Paths
- Median
- …

3SUM

This is where it all started…

3SUM: Given n integers, are there 3 that sum to 0?

-15	-6	33	8	1	-21	4	-30	7	…	107

Naive alg: $O(n^3)$
Simple alg: $O(n^2)$

A famous conjecture in computational geometry:

The 3-SUM Conjecture:
"3-SUM cannot be solved in $O(n^{1.99})$ time."

Best known [BDP '05, GP'14]: $O\left(\frac{n^2}{(\log n / \log \log n)^2}\right)$

Subclasses within P

k-SAT

↓

Diameter
Closest Pair
Local Alignment
Dynamic Reachability
Single-Source Max-Flow
Subtree Isomorphism
Stable Matching
Edit-Distance
Frechet
LCS
…

3SUM

↓ "3SUM-hard class"

Colinearity
Polygon Containment
Strips Cover Rectangle
Triangle Enumeration
Compressed Inner Product
Dynamic Max Matching
Set Intersection
…

3SUM: Given *n* integers, are there 3 that sum to 0?

-15	-6	33	8	1	-21	4	-30	7	…	107

abc-3SUM: Given a set S of *n* integers, are there $x, y, z \in S$ such that $a \cdot x + b \cdot y + c \cdot z = 0$?

(3SUM is 111-3SUM)

<u>For next week:</u>

Try to prove subquadratic-equivalence for all $a, b, c \neq 0$:

"If one is in $O(n^{1.99})$ time then any other is in $O(n^{1.999})$ time"

Subclasses within P

k-SAT

- Diameter
- Closest Pair
- Local Alignment
- Dynamic Reachability
- Single-Source Max-Flow
- Subtree Isomorphism
- Stable Matching
- Edit-Distance
- Frechet
- LCS
- …

3SUM

- Colinearity
- Polygon Containment
- Strips Cover Rectangle
- Triangle Enumeration
- Compressed Inner Product
- Dynamic Max Matching
- Set Intersection
- …

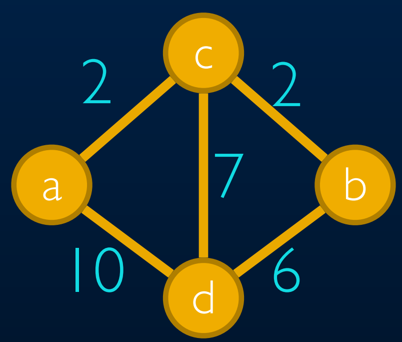
APSP

- Radius
- Dynamic Max Matching
- Stochastic Context-Free Grammar Parsing
- Negative Triangle
- Dynamic Max Flow
- Replacement Paths
- Median
- …

All Pairs Shortest Paths

This is where we saw the full power of fine-grained reductions…

APSP: Given a graph on n nodes and n^2 edges, compute the distance between every pair of nodes.



	a	b	c	d
a	0	4	2	9
b	4	0	2	6
c	2	2	0	7
d	9	6	7	0

[Floyd-Warshall '62] ***O(n³)*** time.

All Pairs Shortest Paths

APSP: Given a graph on n nodes and n^2 edges, compute the distance between every pair of nodes.

Author	Runtime	Year
Fredman	$n^3 \log \log^{1/3} n / \log^{1/3}$	1976
Takaoka	$n^3 \log \log^{1/2} n / \log^{1/2}$	1992
Dobosiewicz	$n^3 / \log^{1/2} n$	1992
Han	$n^3 \log \log^{5/7} n / \log^{5/7}$	2004
Takaoka	$n^3 \log \log^2 n / \log n$	2004
Zwick	$n^3 \log \log^{1/2} n / \log n$	2004
Chan	$n^3 / \log n$	2005
Han	$n^3 \log \log^{5/4} n / \log^{5/4}$	2006
Chan	$n^3 \log \log^3 n / \log^2 n$	2007
Han, Takaoka	$n^3 \log \log n / \log^2 n$	2012
Williams	$n^3 / 2^{\Omega(\sqrt{\log n})}$	2014

Classical Algs: $O(n^3)$

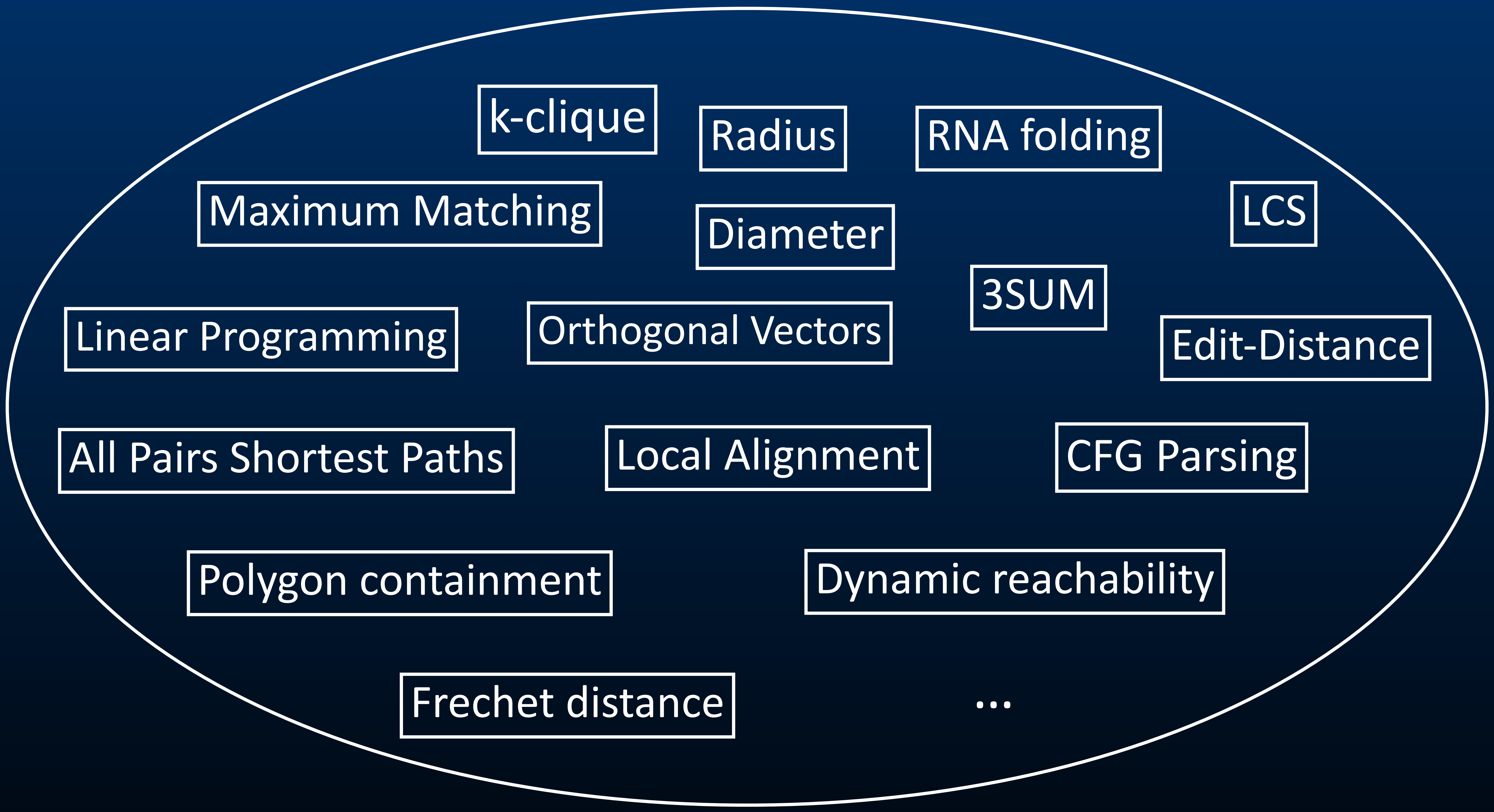
*Floyd-Warshall, n*Dijkstra,…*

Conjecture:

APSP cannot be solved in $O(n^{3-e})$ time.

Many of these are subcubic-equivalent

The Class P (before)



The Class P (after)

k-SAT

- Diameter
- Closest Pair
- Local Alignment
- Dynamic Reachability
- Single-Source Max-Flow
- Subtree Isomorphism
- Stable Matching
- Edit-Distance
- Frechet
- LCS
- …

3SUM

- Colinearity
- Polygon Containment
- Strips Cover Rectangle
- Triangle Enumeration
- Compressed Inner Product
- Dynamic Max Matching
- Set Intersection
- …

APSP

- Radius
- Dynamic Max Matching
- Stochastic Context-Free Grammar Parsing
- Negative Triangle
- Dynamic Max Flow
- Replacement Paths
- Median
- …

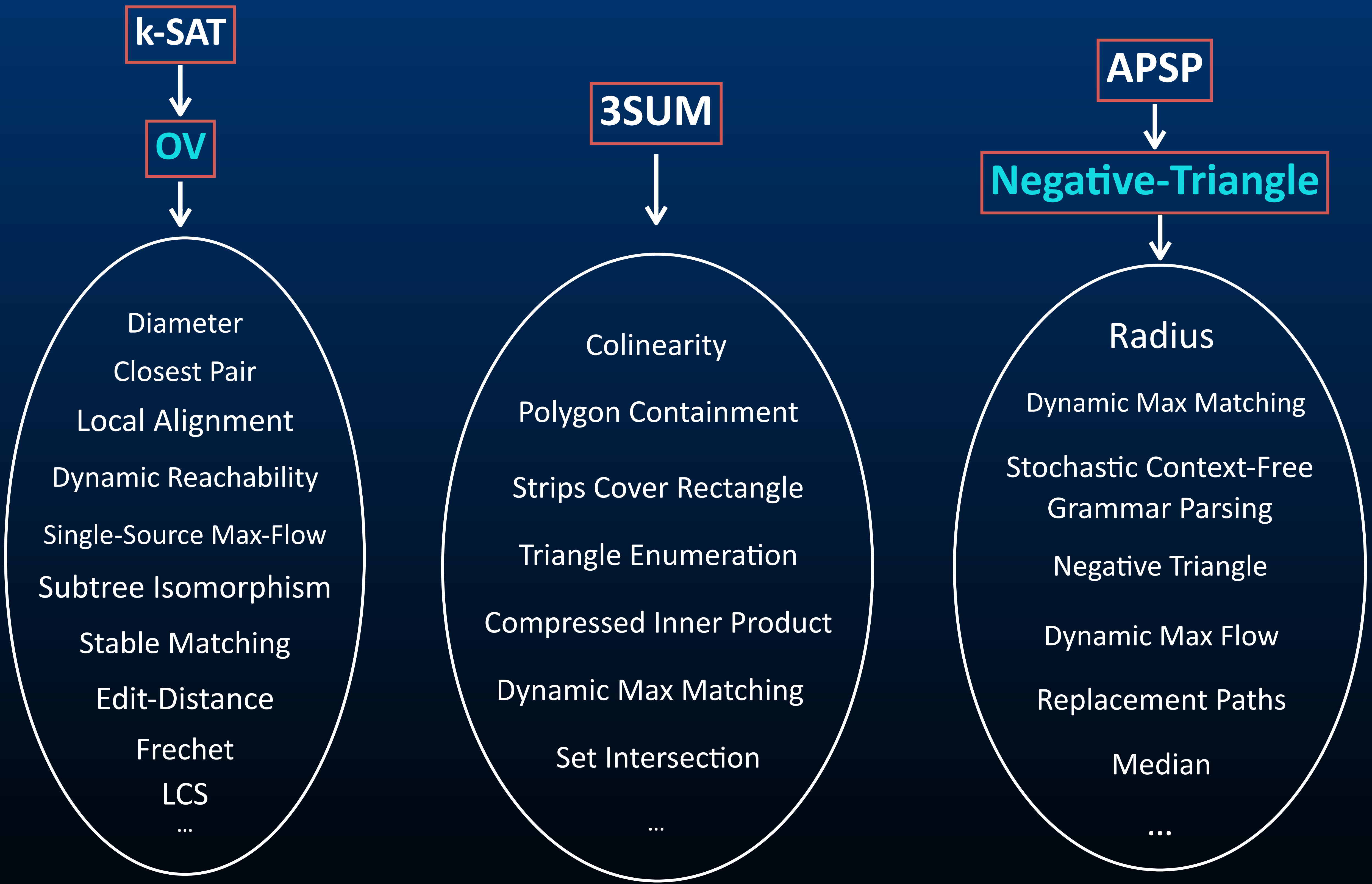
We are starting to understand the structure within P

Today's Lecture

- Motivation
- Intro to Fine-Grained Complexity
- The Basics (Part 1: weeks 2 to 8)
- About this course
- Advanced topics (Part 2: weeks 8 to 14)

Course Objectives and Focus

- Goal 0: The ability to understand FGC results.
- Goal 1: The ability to prove your own FGC results.
  - We will highlight the simplest hard problems (Part 1)


k-SAT
OV
Diameter
Closest Pair
Local Alignment
Dynamic Reachability
Single-Source Max-Flow
Subtree Isomorphism
Stable Matching
Edit-Distance
Frechet
LCS
…
3SUM
Colinearity
Polygon Containment
Strips Cover Rectangle
Triangle Enumeration
Compressed Inner Product
Dynamic Max Matching
Set Intersection
…
APSP
Negative-Triangle
Radius
Dynamic Max Matching
Stochastic Context-Free Grammar Parsing
Negative Triangle
Dynamic Max Flow
Replacement Paths
Median
...

Course Objectives and Focus

- Goal 0: The ability to understand FGC results.
- Goal 1: The ability to prove your own FGC results.
  - We will highlight the simplest hard problems (Part 1)
  - We will see new conjectures and variants (Part 2).
  - Algorithms given for enrichment, and to know the limits.
- Goal 2: Intimacy with the theory and with current research.
  - This is the purpose of the advanced topics (Part 2).
- *Most importantly: To have fun thinking about basic problems!*

Technical Remarks

- We will ignore $\log n$, $\log^{O(1)} n$, $2^{\sqrt{\log n}}$ or any $n^{o(1)}$ factors.
  - Many reductions have such overheads.
- We allow randomness.
  - The conjectures are assumed to holds against randomized algorithms too.
  - Many reductions use randomness.
- We use the (standard) Word RAM model with $w = O(\log n)$.
  - You can do any operations on words in constant time: addition, multiplication, random access, hashing, etc.
  - Since we allow log factors and randomness, this is not too important.
- Numbers are assumed to be in a polynomial range.
  - Integers in $\{-n^{O(1)}, \ldots, +n^{O(1)}\}$, real numbers with precision $1/n^{O(1)}$.

Today's Lecture

- Motivation
- Intro to Fine-Grained Complexity
- The Basics (Part 1: weeks 2 to 8)
- About this course
- Advanced topics (Part 2: weeks 8 to 14)

On The Conjectures

- Why do we need 3+? Can we reduce them to one another?
  - Nonreducibility Results.
  - Some connections and partial unifications.
- What happens if they fail? Can we use more reliable conjectures?
  - Better conjectures and the consequences of breaking them.

[AHVW '16]

Lower bounds under a better "SETH"

k-SAT

Diameter

k-Dominating-Set

Dynamic reachability

Stable Matching

Local Alignment

Edit-Distance

Frechet

LCS

…

Circuit-SAT

Faster LCS → Faster Circuit-SAT → Breakthroughs in Complexity Theory

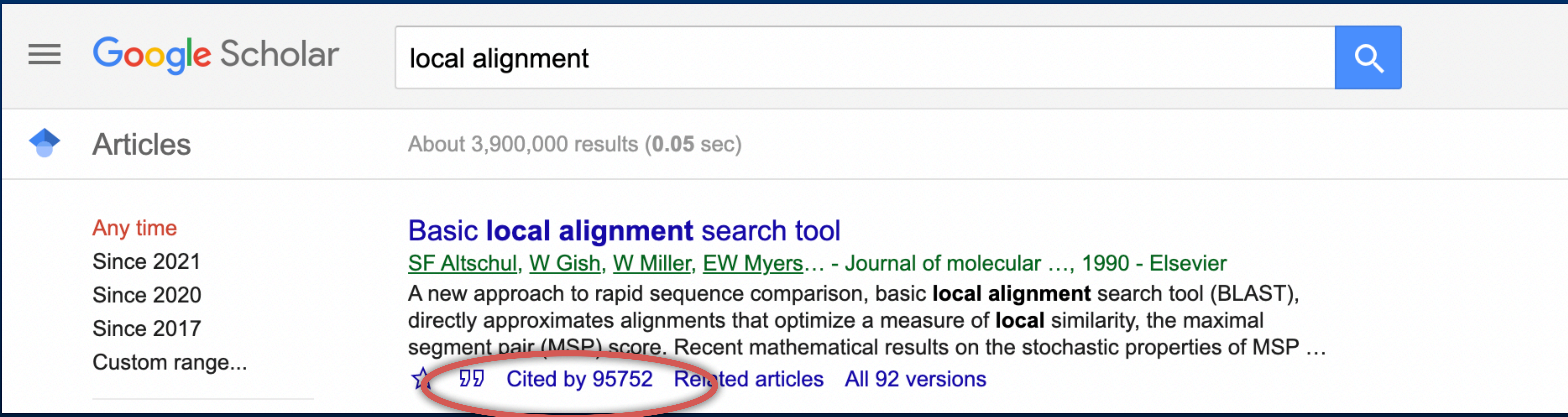
e.g. $O(N^2/\mathit{log}^{100} N)$

breaks crypto one-way functions…

Beyond Worst Case

- Approximations?

BLAST: A *heuristic, linear time* algorithm for **Local Alignment**.



95k citations!

No fast exact algorithm under SETH.

*Are there fast algorithms with **some** optimality guarantees?*

Hardness of Approximation in P

Big open questions:

Best Approximation for LCS and Edit Distance in $O(n^{2-\epsilon})$ time?

Most optimistically:

(1.0001)-approximation in linear time?

After a very long sequence of works [DCGKS'18, AN'20]:

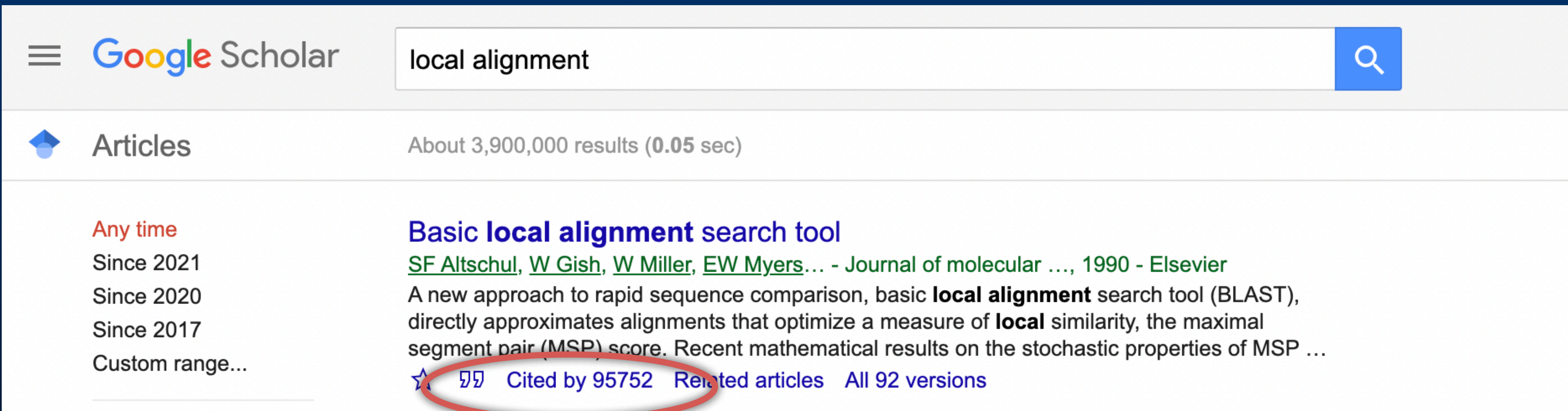
Near-linear time **O(1)** approximation for **Edit Distance.**

Big open question: Prove any 1.00001 lower bound.

Beyond Worst Case

- Approximations?
  - Some techniques for hardness of approximation (including a fine-grained “distributed PCP” result).
- Average-case?
  - Some techniques, with applications in cryptography.
- Restricted inputs?
  - Tight results in terms of multiple parameters.

BLAST: A *heuristic, linear time* algorithm for **Local Alignment**.



95k citations!

No fast exact algorithm under SETH.

What is the parameterized complexity under natural parameters?

Parameterized Complexity in P

[Bringmann-Kunnemann]

parameters for LCS:

$n = \|x\|$	.. length of longer string
$m = \|y\|$	.. length of shorter string
$L = LCS(x, y)$	.. length of LCS
$\|\Sigma\|$	.. size of alphabet Σ
$\Delta = n - L$	.. number of deletions in x
$\delta = m - L$	.. number of deletions in y
M	.. number of *matching pairs*
d	.. number of *dominant pairs*

multivariate algorithms: $\tilde{O}(n + \min\{d, \delta m, \delta\Delta\})$

under SETH, this is **optimal** for any relations $m = \Theta(n^{\alpha_m}), L = \Theta(n^{\alpha_L}), \ldots$

* Slide from Karl Bringmann

Beyond Worst Case

- Approximations?
  - Some techniques for hardness of approximation (including a fine-grained "distributed PCP" result).
- Average-case?
  - Some techniques, with applications in cryptography.
- Restricted inputs?
  - Tight results in terms of multiple parameters.
  - Results for restricted input families.

Beyond (Classical) Time Complexity

- Fine-grained complexity in a quantum world?
  - Grover's search helps but not by too much (Quantum-SETH) and not always.

Quantum Time Complexity via Grover's Search

~~2^n~~ $2^{n/2}$ **k-SAT**

- Diameter
- Closest Pair
- Stable Matching
- Dynamic Reachability
- Single-Source Max-Flow
- Subtree Isomorphism
- Local Alignment
- Edit-Distance
- Frechet
- LCS
- …

still n^2 ?

~~n^2~~ n **3SUM**

- Colinearity
- Polygon Containment
- Strips Cover Rectangle
- Triangle Enumeration
- Compressed Inner Product
- Dynamic Max Matching
- Set Intersection
- …

~~n^3~~ $n^{2.5}$ **APSP**

- Radius
- Dynamic Max Matching
- Stochastic Context-Free Grammar Parsing
- Negative Triangle
- Dynamic Max Flow
- Replacement Paths
- Median
- …

Recap

Popular since 1970's

(Traditional) Complexity:

Polynomial vs. *exponential?*

My problem is in P

$\Downarrow$

$P = NP$ (very unlikely!)

$O(n^c)$ *or* $O(c^n)$ *?*

"Polynomial = efficient"

A theory for Small Data

Popular since 2010's

Fine-Grained Complexity:

Linear vs. super-linear?

My problem is linear

$\Downarrow$

SETH is false (unlikely!)

$O(n), O(n^{1.5}), O(n^2), \ldots ?$

"Near-linear = efficient"

A theory for Big Data

3SUM: Given *n* integers, are there 3 that sum to 0?

-15	-6	33	8	1	-21	4	-30	7	...	107

abc-3SUM: Given a set S of *n* integers, are there $x, y, z \in S$ such that $a \cdot x + b \cdot y + c \cdot z = 0$?

(3SUM is 111-3SUM)

<u>For next week:</u>

Try to prove subquadratic-equivalence for all $a, b, c \neq 0$:

"If one is in $O(n^{1.99})$ time then any other is in $O(n^{1.999})$ time"