SIGACT News Complexity Theory Column 100

Lane A. Hemaspaandra
Dept. of Computer Science
University of Rochester
Rochester, NY 14627, USA

Introduction to Complexity Theory Column 100

This Issue

This is column number 100, and Bill Gasarch has very kindly made it an event! In particular, this issue's column is the third of Bill Gasarch's series of polls on the field's thoughts on P vs. NP (and other central issues in complexity). The first two polls in the series appeared as SIGACT News Complexity Theory Columns 36 and 74.

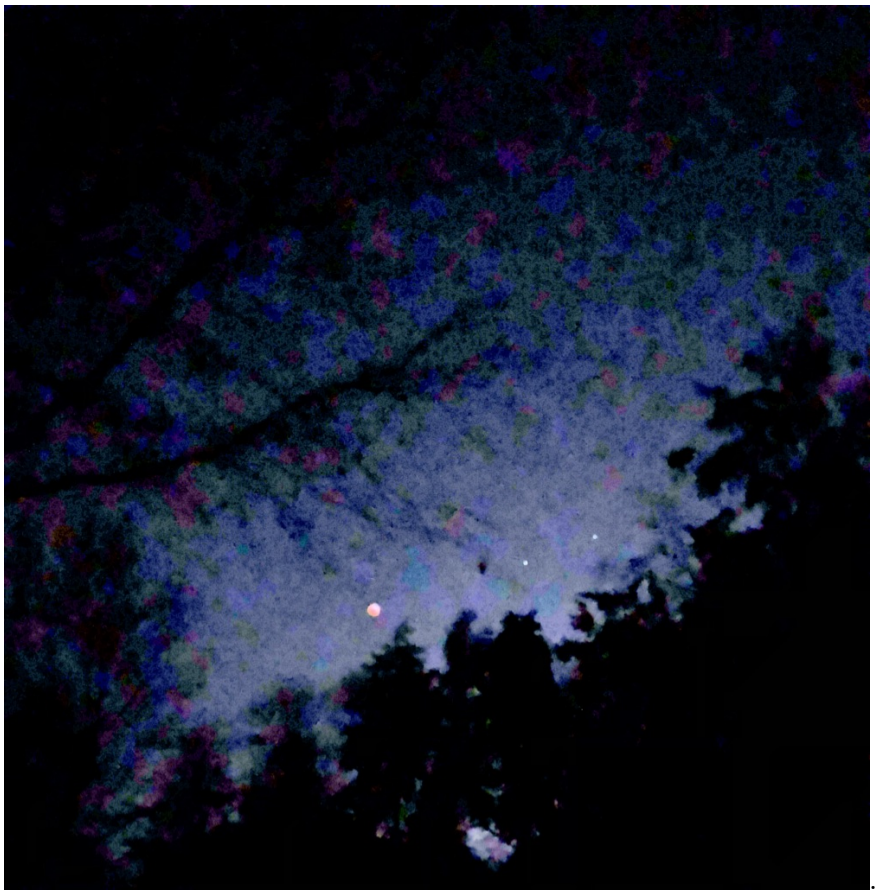
Warmest thanks to Bill Gasarch for having undertaken the huge task of creating the poll, gathering the responses, analyzing them, and writing this article on what the poll reveals, and to Clyde Kruskal who was Bill's magical elf on proofreading, polishing, and even knocking on doors to ask people what they think of P versus NP. (Psst: Bill and Clyde's 2019 book, "Problems with a Point: Exploring Math and Computer Science" (https://www.worldscientific.com/worldscibooks/10.1142/11261), sounds absolutely fascinating!)

Coming Issues

But wait! Surely, one issue of Bill is not enough. So how about two, to make this The Half-Year of Bill! In particular, Column 101 will be by Bill in his nonpollster hat, on the muffin problem. And if even two is not enough for you, let me mention that if you flip, after reading this poll-results article, to Bill's column in this very same issue, you'll find Bill's own answers to the poll's questions!

Bill is a hard act for anyone to follow, but happily he'll be followed by others who are also hard acts to follow. Please stay tuned for future columns here from Emanuele Viola (topic: TBD); from Aviad Rubinstein (topic TBD); and from Sabine Broda, António Machiavelo, Nelma Moreira, and Rogério Reis (tentative title: Average Descriptional Complexity of Regular Languages).

By the way, it is a sure thing that Bill's polls will be at least as frequent, during the twenty-first century, as super wolf blood moon eclipses: After all, this is the third of Bill's polls and there will be exactly three super wolf blood moon eclipses this century; the second one was January 2019

 .

(and is shown above, from Düsseldorf; the pink circle between the trees is the moon in total eclipse) and the third will be in 2037. But I think Bill is going to outdo nature here: Looking into the distant future as to coming columns, note that Bill's Section 4 hints at a likely fourth P =? NP poll by Bill or his designated robot!

Ker-I Ko, 1950–2018

Professor Ker-I Ko, a wonderful, insightful complexity researcher, who also was one of the very best at writing textbooks conveying the gorgeous landscape of what is known, passed away in December.

An *In Memoriam* from Ding-Zhu Du and Jie Wang appears later in this issue, and will cover some of Ker-I's many contributions. Also, the blog post (`https://blog.computationalcomplexity.org/2018/12/ker-i-ko-1950-2018.html`) on Ker-I by Lance Fortnow, after Lance learned of Ker-I's passing, covers some of the highlights of Ker-I's work: on oracles under which the polynomial hierarchy collapses after a specific number of levels, on instance complexity, and on whether 1-1-length-increasing equivalence implies nonisomorphism; and that posting also mentions Ker-I's work in the study of complexity-theoretic aspects of real functions, a

topic to which he was deeply devoted.

But let me please mention that Ker-I's interests were so broad, and his contributions so valuable, that even if one were to set to the side Ker-I's work in all of the results/topics mentioned by Lance, each of which indeed is a very important result/topic, one would find in the rest of Ker-I's output remarkably many other advances that would be among the top career highlights for almost anyone who had made such advances. As just a few examples: Ker-I's 1983 JCSS paper—I believe it was the very first work on semi-feasible computation (P-selectivity theory) after the seminal papers of Alan Selman—showing that all semi-feasible sets have small circuits; Ker-I's influential studies (in a 1988 SICOMP paper with Ron Book and a single-authored 1989 I&C paper) of the sets reducible to sparse sets via various reduction types; Ker-I's 1987 TCS paper on the theory of helping, in which he introduced the notion of strong advice—advice that must work on all lengths up to and including the given one; Ker-I's 1991 I&C paper on the low and high hierarchies, in which he showed that there is a relativized world in which both the low and high hierarchies are infinite—and thus that in that world no set can be both low and high—and that for each k there are relativized worlds where the low and high hierarchies extend exactly k levels; and his 1985 TCS paper that, independently of the paper by Grollmann and Selman that obtained the same result, proved that $\mathrm{P} \neq \mathrm{UP}$ if and only if complexity-theoretic one-way functions exist.

Ker-I, whose brilliance and kindness touched so many people, will be deeply missed.

Guest Column: The Third P =? NP Poll[1]

William I. Gasarch[2]

Abstract

This column summarizes the results from a poll on what people think about P=?NP and related issues.

1 Introduction

In 2001 I (innocently!) asked Lane if I could write an article for his SIGACT News Complexity Theory Column that would be a poll of what computer scientists (and others) thought about P=?NP and related issues. It was to be an objective record of subjective opinions. I asked (by telegraph in those days) over 100 theorists. Exactly 100 responded, which made taking percentages very easy. That poll appeared in the SIGACT News Complexity Theory Column in 2002 (I call it *the 2002 poll* even though people answered it in 2001). The Wikipedia page on P=?NP links to it. That poll's readership and popularity have exceeded my wildest dreams.

In 2011, ten years after the first poll, I conducted another poll. That poll appeared in the SIGACT News Complexity Column in 2012 (I call it *the 2012 poll* even though people answered it in 2011). I had long since disposed of my old telegraph and feather pens. I asked (by email and blegs[3]) over 200 theorists, of which 152 responded.

Since I am a fan of arithmetic sequences of length three, I fully expected to do the next P=?NP poll in 2021, to appear in 2022. So why did I do a poll in 2018, appearing in 2019? Because Notorious LAH asked me if I could do it a bit earlier to coincide with his 100th column.

Before giving you the results and comments, I want to quote myself (who first said *to quote yourself is the height of arrogance*?... either Gauss or Gasarch) from 2012:

> *I purposely did not use Surveymonkey or a similar device since I want people to have the freedom to say things like*
>
> 1. *I hope* P=?NP *is never resolved!, or*
> 2. *my highest (academic) degree: 105 when I was really sick.*

[2] Dept. of Computer Science, University of Maryland, College Park, MD, 20742, USA. gasarch@cs.umd.edu.

[3] A *bleg* is when you use your blog to beg for answers to a question or poll.

For better or worse this time around *I did use* Surveymonkey.

1. I got 124 respondents, which is fewer than last time. Is the drop in respondents because I used Surveymonkey? People are busy? People are distracted by Facebook, Twitter, etc. These are the questions that try one's soul.

2. There were fewer unusual answers. I think this is because in the 2002 and 2012 polls I would ask (for example):

   Is Graph Isomorphism in P?

   People would email me their answers, so they were quite free to write something unusual. By contrast this year I would ask:

   Is Graph Isomorphism in P?

   - YES
   - NO

   I suspect people felt obligated to play by the rules and check off one of them, despite the fact that there was a mechanism to make comments in Surveymonkey.

3. In the 2002 and 2012 polls some questions had fewer than 20 responses. This time every question had at least 80 responses. I think this is for the same reason as the last item: most people felt obligated to give *some* answer, even though they could leave an answer blank.

I summarize the results and compare them to the results from 2002 and 2012. I also list some people's comments.

I do not give my answers to the poll here. For those see my *Open Problems Column* in this issue of SIGACT News (Volume 50, Number 1, March 2019). I do give some personal comments in square brackets [like this].

I refer to this poll as *the 2019 poll* since this article appears in 2019, although the poll was conducted in 2018. The percentages that I give are approximate. When I list comments I do it in reverse alphabetical order of the last name of the commenter to counter the alpha-prejudice of our society.

Section 2 summarizes what *all* of the people who answered the poll said. What about people who are known to have seriously thought about the problem? I discuss their opinions briefly in Section 3.

2 Summary of Results

2.1 Does P=NP?

The following is a chart of the responses to the question **Does P=NP?** in my 2002, 2012, and 2019 polls. DK stands for Don't Know, DC stands for Don't Care, and Ind stands for Independent (which I assume means Independent of ZFC).

	P≠NP	P=NP	Ind	DC	DK	DK and DC	other
2002	61 (61%)	9 (9%)	4 (4%)	1 (1%)	22 (22%)	0 (0%)	3 (3%)
2012	126 (83%)	12 (9%)	5 (3%)	5 (3%)	1 (0.66%)	1 (0.66%)	1 (0.66%)
2019	109 (88%)	15 (12%)	0	0	0	0	0

In 2012 there were many more P≠NP proponents than in 2002. In 2019 there were just a few more. In 2019 everyone was either EQUAL or NOT-EQUAL. Gone are the clever people who say *Ind of ZFC!* or *I don't know and I don't care!* I do not know (though I do care) if the lack of unusual answers is due to a maturing of the field or to the fact that Surveymonkey didn't quite allow for other answers. A few of the comments in 2019 looked like they wanted to say IND, DK, or DC, but not many. I really do think the field has matured, and so have the responders. Of course, saying that voting IND, DK, or DC is immature reveals my bias, which could be wrong.

Dmytro Taranovsky points out that we really don't know:

NO. While I agree with the consensus view, there is a substantial uncertainty on the issue: We fundamentally do not understand computation. Natural classes tend to be comparable and we cannot rule out that this applies to complexity classes as well.

András Salamon thinks P≠NP but they are nearly equal:

NO. My working hypothesis is that NL≠NP *[*NL *is Nondeterministic Log Space] with a lower probability then* P≠NP*[I prefer stating the converse:* P=NP *is more likely than* NL=NP*.] However, it seems clear that finding the hard regions is itself hard, so I aim to spend one in every ten working days exploring the consequences of the hypothesis that* P=NP*, since we seem to live in a world where* P *is "nearly equal to"* NP *for practical purposes (not because problems in* NP *are "easy," but because* P *contains really hard problems by the Deterministic Time Hierarchy theorem).*

Lance Fortnow is more sure that P≠NP:

NO. People that think P=NP *are like people who think Elvis is still alive. [Maybe Elvis will prove* P=NP*.]*

2.2 When Will P=?NP Be Resolved?

The following is a chart of the responses to the question **When Will P=?NP Be Resolved?** The years are 2000+ unless it is written in full. So 02–09 means 2002–2009 but 2200–3000 means 2200–3000.

	02–09	10–19	20–29	30–39	40–49	50–59	60–69	70–79
2002	5 (5%)	12 (12%)	13 (13%)	10 (10%)	5 (5%)	12 (12%)	4 (4%)	0 (0%)
2012	0 (0%)	2 (1%)	17 (11%)	18 (12%)	5 (3%)	10 (6.5%)	10 (6.5%)	9 (6%)
2019	0 (0%)	0 (0%)	26 (22%)	20 (17%)	14 (12%)	9 (7%)	7 (6%)	5 (4%)

	80–89	90–99	100–109	110–119	150–159	2200–3000	4000–4100
2002	1 (1%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	5 (5%)	0 (0%)
2012	4 (3%)	5 (3%)	2 (1.2%)	5 (3%)	2 (1.2%)	3 (2%)	3 (2%)
2019	0 (0%)	0 (0%)	1 (0.8%)	10 (12%)	10 (12%)	1 (0.8%)	11 (9%)

	Long Time	Never	Don't Know	Sooner than 2100	Later than 2100
2002	0 (0%)	5 (5%)	21 (21%)	62 (62%)	17 (17%)
2012	22 (14%)	5 (3%)	8 (5%)	81 (53%)	63 (41%)
2019	7 (6%)	11 (9%)	0 (0%)	84 (66%)	40 (34%)

In 2002, 62% thought that P=?NP would be resolved by 2100; however, in 2012 only 53% felt that way. In 2019 it is up to 66% which amazes me because, since 2012, there has been little (no?) progress on resolving P=?NP. Note that in 2019 more people thought it would never be solved than in 2012 or 2002.

Here are some comments I found amusing XOR insightful:

Peter Shor:

2050–2059. I think nobody can see more than 30 years in the future, so all this says is that I don't think it will be before then.

Benoit Razet

I'd hope in the next 50 years. I'd really like to see it in my lifetime since it's going to be such a breakthrough whatever the answer is. And it will impact other big questions in computational complexity.

Clyde Kruskal:

I am sticking with my answer on the original poll (2036), so that I can be wrong only once. Who knows, maybe I will be right.

Ryan Krusinga:

It will be resolved sooner than people think due to advancements in artificial intelligence. Even if full AGI (artificial general intelligence) is difficult or impractical, artificial mathematicians should still become good enough to assist in advanced proofs to resolve P=?NP*. [See Bogdan Grechuk's comment for a similar yet different answer to this question.]*

William Hoza:

The P=?NP *problem will never be resolved. I'm pessimistic that anyone will ever prove* P≠NP*, but I'm optimistic that we will someday prove that there is no proof that* P≠NP*.*

Bogdan Grechuk:

I think P=?NP *will be resolved when most of the open questions in mathematics will be resolved by using a quantum computer. I am not sure if we should call such a period "a golden age" or, conversely, "the death of mathematics". [See Ryan Krusinga's comment for a similar yet different answer to this question.]*

Hal Gabow:

2018–2029. There are too many smart people working on this for the problem to remain unsolved. [Does the same reasoning imply that the Riemann Hypothesis will be solved soon?]

Lance Fortnow:

It's a Poisson distribution so I'll always say 50 years in the future.

Anonymous:

The day after I die. And due to that answer, I'll answer the question about printing my name "No", since otherwise, if anyone is so utterly unwise as to believe my "when" answer, it will be open season on me . . . eek! [On the one hand, I have respected your wishes and not revealed your name. On the other hand, I know your name. And I really want to know the answer. . .]

And of course there is always some **Anonymous Wiseguy:**

When I solve it!

2.3 If P=?NP Were Resolved Tomorrow Then ...

Most people hold the following views at the same time:

1. P≠NP.
2. If P=NP then this could be known tomorrow – some algorithmic trick or math theorem (think Graph Minor Theorem) just pops and you have a polynomial time algorithm for SAT!
3. If P≠NP then we are in for the long haul. There has been no progress. When I tell my friends and family that we have made progress on ruling out techniques they just laugh at me. :-(

These beliefs are not contradictory. But they do allow for a question where you have to choose between *I think* P≠NP and *I think it will be a very long time before we prove* P≠NP. Here is the question:

(Only answer this question if you believe that P≠NP*.) Sasha Razborov, Avi Wigderson, and Andy Yao (or three other wise sages whose opinions on* P=?NP *you take seriously) knock on your door at 3:00AM to tell you that* P=?NP *has been resolved – but after announcing it dash off to tell Lane the good news – without telling you in which direction or how it had been resolved!* ***Which way do you think it went?*** *(This question measures what is stronger: your belief that* P≠NP *or your belief that we are nowhere near proving* P≠NP*.)*

89 responses.

- P≠NP: 72 (80%)
- P=NP: 17 (20%)

I was surprised that only 17 (20%) would switch to P=NP. I know I would. I present some comments, the first one of which may explain why people largely stayed with P≠NP. This may be because the three wise sages all think P≠NP (several of the respondents made that point). If I ask this question again on my next poll I may make it a group with people on both sides of the issue. One of my proofreaders, David Sekora, points out that then the question would measure which wise sages you regard more highly. Darn!

András Salamon:

If Moshe Vardi, Donald Knuth, Richard Lipton, and Emanuele Viola [serious people who think P=NP*] had been at the door, then I'd have gone* P=NP *:-). I don't think this question necessarily measures what you intended.*

Kelly J Rose:

I would think P=NP *because that's the only reason they would bug me at 3:00AM.*

Jacob Kesinger:

I don't believe any results after midnight.

Lance Fortnow:

Since these are complexity people they will have proved P≠NP*. Now if it was Richard Karp, Robert Tarjan and Noga Alon [people who work on algorithms] that might be a different story.*

Stephen Fenner:

Do they bring gold, frankincense, and myrrh with them?

Jeremy Alm:

Independent of ZFC.

2.4 What Kind of Mathematics Will Be Used to Resolve P=?NP?

The number of people who ventured a guess as to what techniques will be used is 93 (up from 71 in 2012).

About 40 think it will require new or hard math. By contrast 63 thought so in 2012. Of course, it's hard to know if someone who thinks it will require (say) Integer factoring and Integer Programming thinks it will require new or hard math.

GCT (geometric complexity theory) was alluded to by 8 people. Back in 2012 it was mentioned explicitly by 11.

We list all of the answers given. Answers listed below in the form

Recursive Algebraic Topology – 13

means that 13 people gave that response.

I have put the answers into categories: *Old School*, *GCT*, *Left Field*, *Several Different Fields*, *Predictions of More than Techniques*, and *The Wisest in the Land*. The categories are not perfect. Some may argue that I put answers in the wrong category. Oh well.

Within a category I list answers in order of which one got the most votes. In case of a tie I first list answers that had a person attached to them (the people in reverse alphabetical order) then the other answers (in reverse alphabetical order). If someone had a long answer I give the identity of who said it.

Old School. These are answers that rely on computer science that was known 20 years ago. Who knows? They could be right. Maybe the key does lie in tightening up the relation between CFG's and PDA's.

1. Logic. – 3
2. Combinatorics. – 3
3. Set Theory. – 2

The rest of the *Old School* answers got one vote each.

1. **Thomas Klimpel:** *By methods related to the Berman–Hartmanis conjecture. We will first learn how to separate lower complexity classes by both really knowing why they are different and expressing this knowledge in terms of appropriate isomorphisms between the problems really capturing the essence of the complexity class. And after we learn how to prove separations, and hence understand why two classes are different, somebody will brute force this missing knowledge.*

2. **Anonymous** *Magical algebraic techniques proving astonishingly powerful circuit lower bound. [Reminds me of Barrington's result that* $\mathrm{NC}^1 \subseteq \mathrm{BP}^5$*; however, note that this is an upper bound.]*

3. **Jeremy Alm:** *Whatever Saharon Shelah decides to invent for the purpose! [Shelah has been active for a very long time so this is an Old School answer. I am not sure if it's a serious answer. Then again, some people think it was not a serious question. So fair is fair.]*

4. Turing Machines.

5. Model Theory and Diagonalization.

6. Counting.

7. Bounded Arithmetic.

8. Boolean Logic Circuits.

9. Algebraic Complexity Theory.

GCT: None of the answers mentioned *Geometric Complexity Theory (GCT)* directly; however, several allude to the math that GCT entails and I suspect they were thinking along those lines.

1. Algebraic Geometry. – 6

2. Weird Geometry. – 1

3. Not Algebraic Geometry alone. – 1

Left Field: Some field of math not usually associated with complexity theory.

1. Continuous methods. – 3

2. **Ken Regan:** *Higher Cohomology [When I asked this in the 2002 poll Ken Regan wrote* Higher cohomology is inevitable*. Glad to see he's sticking to that. Maybe he's right.]* – 1

3. Nonconstructive. – 1

4. Group Theory. – 1

5. Not Knot Theory. – 1

Several Different Fields: Why pick one field of math when you can pick two, or three, or ...

1. Algebra and Complexity Theory. – 4

2. Combinatorics, Probability, and Algebra. – 3

3. Combinatorics, Physics, and Neural Nets. – 2 [Really! Two people picked these three fields? What are the odds of that? To answer that I need to add Probability to the list!]

The rest had one vote each.

1. **András Salamon:** *Combinatorics, but possibly also some (universal) algebra and finite model theory. Or whatever tools someone like Benjamin Rossman has in their toolkit at the time. [All the cool kids go with the Ross-man!]*

2. **Benoit Razet** *Not sure, it could be plain complexity theory, theoretical computer science, or maybe more pure mathematics, combinatorics, probabilities, and maybe linear programming.*

3. **Damiano Mazza:** *A grand reunification of Theory A and Theory B, i.e., something that goes through a more compositional, e.g., type-theoretic, category-theoretic, etc., approach to the combinatorics of computation and complexity.*

4. **Mitch Harris:** *The first proof will involve a lot of machinery from algebraic geometry and analytic number theory, but eventually those methods will be removable from the proof. We won't be left with a simple proof, but what is left will be mostly logical (a mix of model theory and proof theory). Of course this is only speculation.*

5. **Joshua Grochow:** *All kinds: algebraic geometry, combinatorics, group theory, number theory, representation theory, analysis, and probably math that hasn't been invented yet. If I had to bet on just one, it'd be algebraic, or perhaps arithmetic geometry (but maybe that's cheating because it can kind of combine all the others).*

6. **Anonymous:** *Newly developed kind of math that unites seemingly completely different areas such as Ramsey theory, SDP hierarchy, ML, and dynamical system. I have completely no idea what I'm talking about LOL...*

7. **Eric Allender:** *What does this question even mean? I think that the answer is NOT "logic" or "algebraic geometry" or "automata theory" or any other traditional branch of mathematics. But the tools will have evolved from "standard" branches of mathematics, and it will still be "mathematics" of some kind.*

8. Stochastic Manifolds. [Is that really a thing?]

9. Logic, Number Theory, Algorithms, and Algebraic Topology.

10. A combination of Group-Theoretic Set Theory and Set-Theoretic Group Theory.

11. Arithmetics, Boolean Algebra, and Representation Theory.

12. Algebra and Set Theory.

Predictions of More than Techniques:

1. **Lane Hemaspaandra:** *It depends on which direction it gets resolved. And if it resolves as* P=NP*, it could even be resolved by some arXiv.org P-time at-first-crackpot-seeming-algorithm-for-an-NP-complete-problem posting that turns out to not be crackpot at all. Wouldn't that be a hoot!*

2. **Oliver Friedman:** *Something completely incomprehensible. The proof will be discovered by a neural net proof assistant. Humans will be able to verify that every single step of the proof is correct, but will not have any intuition as to why it is true.*

The Wisest in the Land [with my wise comments]:

1. New Math. [I don't think Tom Lehrer's song *New Math* will help much.] – 16
2. No idea. [The most honest in the land!] – 8
3. Hard Math. [I would think so!] – 5
4. If I knew I would just use it. [Same here!] – 2
5. **Dmytro Taranovsky:** *A fundamental breakthrough in theory. [A tautology: Solving the most fundamental problem in theory will be a fundamental breakthrough in theory.]* – 1
6. Proof by contradiction. [Yikes! Intuitionists won't accept it! All five of them!] – 1

Misc:

1. Integer Factoring and Integer Programming. – 2
2. **Subramaniyan N:** *A practical usable sub exponential algorithm [Ah, an optimist! And Subramaniyan also thinks that it will be solved by 2029. The world needs optimists!].* – 1
3. **Anonymous:** *If* P=NP *then some nonconstructive method. If* P≠NP *then exploiting connections between different fields of mathematics that have been established by complexity theory.* – 1

2.5 Does the Polynomial Hierarchy Collapse?

103 responses.

- YES: 15 (15%)
- NO: 88 (85%)

I should have asked this just of the people who thought P≠NP! However, I didn't have to since I have all the data nicely organized and can just remove those people.

Responses from people who think P≠NP.

93 responses.

- YES: 7 (7.5%)
- NO: 86 (92.5%)

Two people who thought P=NP also thought that the Polynomial Hierarchy did not collapse. I think they were confused.

None of the YES's left a comment. All of the comments from the NO's expressed a lack of confidence.

Lane Hemaspaandra:
Gut = PH does not collapse. But intuition peters out with the increasing of the levels. It would be great if we had an analogue, for polynomial hierarchy levels, of the automatic complete-problem level-boosting work that Hartmanis and Lewis did in JCSS 1971 for the arithmetical hierarchy. [The comment is referring to The Use of Lists in the Study of Undecidable Problems in Automata Theory, by Juris Hartmanis and Forbes Lewis, JCSS, Vol 5, No 1, pages 54–66, 1971.*]*

2.6 Does SAT have Polynomial-Sized Circuits?

99 responses.

- YES: 9 (9.1%)
- NO: 90 (90.9%)

Responses from people who think P≠NP.

91 responses.

- YES: 3 (3.2%)
- NO: 88 (96.7%)

Two people who thought P=NP also thought that SAT did not have polynomial-sized circuits. I think they were confused.

The NO's left many comments, all of which indicate a lack of confidence in their answer.

Anonymous:
NO, but YES for a large number of instances, especially if they are correlated.

Many commenters, paraphrased:
NO, but with far less confidence than the belief that P≠NP*.*

2.7 Does P=BPP?

94 responses.

- YES: 61 (65%)
- NO: 33 (35%)

Back in 2012 only 13 people answered. All 13 voted P=BPP. This year far more answered, and far more thought P≠BPP. Why did so many more people have an opinion? Was that because of Surveymonkey? Why do so many think P≠BPP? None of my friends do! One conjecture about the change is that the stream of results on derandomization seems to have slowed down. For example L=RL (Randomized Log Space) seems harder to prove now than it did ten years ago. Maybe it's not true! Another conjecture is that a higher percentage of non-experts answered. (The results in Section 3 indicate that experts largely believe P=BPP.)

András Salamon:

NO. Derandomization is neat when it works, but I see no reason for it to have magical powers. Notwithstanding Impagliazzo–Wigderson, I believe subexponential-sized circuits are sufficient for E.

2.8 Does SAT∈BQP Imply That the Polynomial Hierarchy Collapses?

80 responses.

- YES: 40 (50%)
- NO: 40 (50%)

The comments were mostly either (1) don't know, (2) since SAT $\notin$ BQP the answer is yes but that's as hard as what you are asking, and (3) random.

Peter Shor:

YES. Since SAT *is not contained in* BQP*, the answer is obviously "yes". But I think it's quite possible you'll only prove it by showing* SAT *is not in* BQP*.*

Lane Hemaspaandra:

Scott Aaronson has my proxy on this question. I traded it to him during a phone call in exchange for "$0.01 and all real estate assets Scott Aaronson currently owns in the Commonwealth of Massachusetts." It was only after the phone call that I looked at the caller ID: Texas. DOH!

Scott Aaronson:

YES. But only for the tautological reason that I don't think SAT is in BQP, and a falsehood implies anything.

2.9 Does P=NP∩co-NP?

98 responses.

- YES: 27 (28%)
- NO: 71 (72%)

Responses from people who think P≠NP.

89 responses.

- YES: 19 (21%)
- NO: 70 (79%)

One person who thought P=NP also thought that P≠NP∩co-NP. I think ze[4] was confused.

Paul Beame:
NO. Factoring and Nash equilibria for general games are in the intersection but both seem highly unlikely to be in P.

Lane Hemaspaandra:
NO. But due to recursive function theory (r.f.t.) the universe wants the answer to be Yes. The universe also, of course, similarly driven by what holds in r.f.t., wants all NP*-complete sets to be p-time isomorphic, despite the probability 1 result to the contrary by Kurtz-Mahaney-Royer. It is interesting that* P *versus* NP ∩ co-NP *itself is, even after all these years, still open as to its status relative to a random oracle. [Using rft (all the cool kids use rft instead of r.f.t., just like all the cool kids dig Notorious LAH not Notorious L.A.H.) as a guide to complexity theory is so 1990's. Calling the field rft instead of computability theory is also so 1990's. Now watch, rft will be used to resolve* P=?NP *and I will be forced to eat humble pie.*[5]*]*[6]

Many commenters, paraphrased:
NO (YES), but with far less confidence than the belief that P≠NP*.*

2.10 Is Graph Isomorphism in Polynomial Time?

108 Responses

- YES: 52 (48%)
- NO: 56 (52%)

Responses from people who think P≠NP.

98 Responses

- YES: 42 (43%)
- NO: 56 (57%)

[4]The word *ze* is a candidate for a genderless pronoun.

[5]I have only ever heard the expressions "eat humble pie" and "eat crow" on TV shows.

[6][[But, Notorious WIG, forgetting who is downstream of you in your article's publication flow is soooooo 2010s! Don't you remember all the side comments editor Stan "The Man" Lee put into comic books back when we were young? So... rft is not a perfect guide to complexity; P vs. NP∩coNP, the isomorphism conjecture for NP, and whether NP is P-immune are all cases where the rft analogues likely are poor guides. But, myself, I hope and expect that the way that rft's structures and results have framed and guided complexity (reductions; classes and completeness; NP and the polynomial hierarchy; immunity; semi-feasibility; degrees; and so much more) is not just in our past but also is part of our future as a field. In fact, if that is not a part of the tool kit of the cool kids of the 2030s, please remember to send me a slice of that pie, Notorious WIG! – Signed, Innocuous LAH]]

In 2012, of the 21 people who commented on it, 14 thought GI∈P, 6 thought GI∉P, 2 thought GI will take $n^{O(\log n)}$ time, which I count as not in P.

Babai's result (Graph Isomorphism is in time $2^{(\log n)^{O(1)}}$, which is called *quasipolynomial time*) could inspire a YES or a NO:

- Babai got GI into quasipolynomial time! It's just a matter of quasipolynomial time before the problem is in quasi-quasipolynomial time! And eventually in quasi$^{\omega}$polynomial time which we all know is P.
- Babai and others have said that current techniques cannot do any better than quasipolynomial time. So Babai's result could also be the lower bound. Perhaps we can formalize this thought and get a proof that GI is not in P, though that would prove P≠NP. So either (a) we shouldn't try, or (b) we really, really, really should try!

Fred Green thinks Babai's result will be the starting point in proving GI in P. He also notes his own change-of-mind on the issue.

This is probably the one opinion of mine that has significantly changed since you last ran this poll. In 2012 I thought that GI *in* BQP *was pretty far off. [It may still be far off; however, I get his meaning – in 2012 he thought* GI *was hard.] Thanks to Babai's efforts, I now think* GI *in* P *has a chance of being proved in the near future.*

2.11 Is Factoring in Polynomial Time?

108 Responses (Yes, the same number as answered the Graph Isomorphism Question!)

- YES: 38 (35%)
- NO: 70 (65%)

Responses from people who think P≠NP.

98 Responses

- YES: 28 (28.5%)
- NO: 70 (71.5%)

In 2012, of the 21 people who commented on it, 8 thought factoring is in P, 13 thought factoring is not in P. In 2019 many more people thought factoring is not in P. In *The Joy of Factoring*, by Samuel Wagstaff, he points out that there has been no progress on factoring since 1985 and states some obstacles to progress (he means provably better algorithms). Perhaps we are better informed on how hard factoring is now than we were in 2012.

In 2012 ten participants speculated that the NSA or the KGB or some other organization might have gotten factoring in P and not told anyone. In 2019 zero participants made that speculation. Why? Perhaps there are enough academics working openly on factoring that (people think) if current mathematics suffices to show factoring is in P, then one of them would have found it. Or perhaps those 10 have disappeared mysteriously.

In 2019 many comments were about how unsure they were of their opinion. We give a sample of other comments.

Tom Wong gives a reason why Factoring is probably not in P, but not a math reason:

So much effort has gone into researching factoring, that I think it's unlikely that a polynomial time algorithm exists.

Peter Shor notes that there are no obstacles to factoring in P:

I don't really see why it shouldn't be.

Mitch Harris reminds us of the contrast between theory and practice:

But it's probably not terribly super-polynomial. $n^{\log \log n}$ *that kind of thing.*

Anonymous

I think we live in Cryptomania! [If you don't know what this means then Google Impagliazzo's Five Worlds.*]*

And of course there is always some **Anonymous Wiseguy:**

Yes factoring is in P*. But I'm busy using my algorithm to break computer security and rake in billions of dollars. [This is a different Anonymous Wiseguy than for Question 2 about when* P=?NP *will be resolved.]*

2.12 If You Answered P≠NP Above, Do You Believe that an Obstacle is Hard Instances?

As an example of the question, let Turing Machine M accept the language

$$L = \{(N, x, 1^t) \mid \text{nondeterministic machine } N \text{ does not halt on input } x \text{ within t steps}\}.$$

Does there exist $\langle N', x' \rangle$ such that the runtime of M on $(N', x', 1^t)$ is not bounded by some polynomial in t?

70 responses.

- YES: 45 (64%)
- NO: 25 (36%)

Thomas Klimpel

There is of course an abstraction structure in NP*. You can embed one problem into another problem, and that problem again into another problem, and so on. And of course, an algorithm would have a hard time undoing those abstractions. But if you generate hard problems that way, then you always have a proof both ways, i.e., those problem instances come from* NP∩co-NP*. Similarly, you can have problems where other natural hierarchical chains occur, like remotely steering a steering wheel. Those instances easily kill individual algorithms, but they don't separate the complexity classes.*

Anonymous

Every NP-complete problem that I know of has many easy instances. Hence it has to be the few hard instances that make it hard.

2.13 If Someone Shows P=NP, Will This Have a Big Effect on Practical Computing?

118 responses.

- YES: 68 (58%)
- NO: 50 (42%)

Surveymonkey only allowed for a YES or NO vote; however, some of the comments were for an OTHER vote. Hence we give YES, NO, and OTHER comments.

YES:

Dmytro Taranovsky:

While it is possible the solution will be ineffective, the consequences of a fully effective P=NP *would be enormous. It can lead to human immortality in 5 years, or if held secret by a power-seeking group, world government in 2 years.*

Peter Gerdes:

Indirectly, the proof will inevitably involve powerful ideas that will have an effect.

John Tromp:

Crypto will be all but dead. [Contrast this to Mitch Harris' NO answer.]

Scott Aaronson:

The practical impact would come not from the result itself, but from the new ideas needed to achieve it.

NO:

Richard Lorentz:

Probably not. I might be wrong but, e.g., I don't think putting linear programming in P *really had much of a practical effect.*

Clyde Kruskal:

There will probably be something special about NP*-complete problems that still makes them hard to solve.*

Lenwood Heath:

I believe that the problems that we have been kicking around for years as NP*-hard will still be hard to solve in some theoretically describable sense.*

Mitch Harris:

Only a small effect. The constants won't be huge, but physical limits to Moore's law will mean the cross over point is pretty impractical. Not galactic [Lipton and Regan in a Blog Post coined "Galactic" to mean an algorithm in poly time but you would never actually run it either due to large degree or large constants] but let's say interplanetary. Also the algorithms would be extremely

non-trivial. As for cryptography, there will still be hard problems with one-way functions, just at the next higher level in the hierarchy. [Contrast with John Tromp's YES answer.]

OTHER:
András Salamon:

If someone produces an algorithm that decides SAT *in quadratic time, yes (because we already have efficient reductions to* SAT *for many problems of interest). If someone gives a nonconstructive proof, or one with a polynomial with degree that depends on the cardinality of some large finite group, not so much.*

Ryan Krusinga:

Some problems may just have ridiculously impractical polynomial-time solutions, even in the best case. Maybe there will be some creative algorithms that work some of the time, but I don't think most problems will be affected much.

2.14 If Someone Shows P≠NP Will This Have a Big Effect on Practical Computing?

116 responses.

- YES: 22 (19%)
- NO: 94 (81%)

Dmytro Taranovsky thinks yes:

Given enough time, fundamental breakthroughs tend to have a big practical impact.

Peter Gerdes thinks yes:

Well the proof won't but the fact that it's true will.

Hal Gabow thinks not:

We already have put our faith in P≠NP .

2.15 Given That SAT Solvers are Now Quite Good, is P=?NP Still Relevant?

112 responses.

- YES: 106 (95%)
- NO: 6 (5%)

These numbers are about the same as in 2012. In fact, in 2012 it was exactly 106 who said YES, 15 said NO, 3 said Hmmm.

This year many of the comments challenged the premise that SAT Solvers are now quite good. In the 2012 poll nobody made that comment. Why the change? SAT Solves are actually *better*

now than they were in 2002; however, perhaps we ask them to work no harder problems and we are closer to seeing their limitations. I'll be curious how this one looks in my next poll.

Dmytro Taranovsky:
Current SAT solvers may be good on random SAT *problems, but not on the problems we care about.*

William Hoza
SAT solvers aren't "quite good" at, e.g., breaking RSA.

Paul Beame:
CDCL SAT solvers are remarkably successful but also quite brittle. They even perform poorly on practical examples where we know that efficient proofs (even of the right form) exist. There is a natural tendency to focus on the (sometimes surprising) successes of SAT solvers, but there are lots of natural and useful examples where we can prove that current CDCL SAT solvers don't have a chance of succeeding. Ditto for other kinds of SAT solvers. Difficult examples that don't distinguish between solvers tend to get pruned from test sets for competitions which is part of the reason that solvers look so good.

Scott Aaronson:
SAT solvers, impressive as they are, don't look poised to prove the Riemann hypothesis or break Bitcoin anytime soon.

2.16 Aside From P=?NP, Which Other Open Problem do You Most Want to See Solved?

This question has too many diverse answers to summarize them. I list some of the answers and I put them into groups. Within a group I sort by how many people had that answer.

Theoretical Computer Science

- How does BQP relate to BPP? Other classes? – 8
- P=BPP? – 3
- NP=co-NP? – 2
- NC=P? – 2
- Unique Games Conjecture. – 2
- Does multiplication of two n bits number requires superlinear time? – 2

All of these got one vote.

- NP=PSPACE?
- PSPACE=EXP?
- Get better [exponential?] lower bounds on the size of refutations in extended Frege.

- Is the quantum PCP theorem true?
- Is there an algorithm for linear programming that solves a problem with L bits of representation in $O(L)$ time?
- The Sum of square roots problem.
- Anything that involves reductions to SAT.
- Are there concrete tests for pseudorandomness?
- NL = NP?
- DSPACE(n)=NSPACE(n)?
- Is there an i such that $\Sigma_i^p = \Pi_i^p$?
- ALOGTIME = RL?
- Is there an efficient algorithm for semi-definite programming?
- Resolve all relations between complexity classes related to Fixed Parameter Tractable and to Fine-Grained Complexity.
- EXPTIME=PSPACE?
- Is Graph Isomorphism P-hard? [B is P-hard if for every $A \in$ P, $A \leq B$ with uniform NC-reductions.]
- What is the complexity of Boolean Matrix Transpose on a multitape Turing machine?

Mathematics

- Riemann Hypothesis. – 4
- All other Clay problems. – 1
- Consistency of New Foundations [New Foundations is a set theory where sets need not be well founded.] relative to ZFC. – 1
- Goldbach's Conjecture. – 1
- Prove or disprove *The Erdős–Turán Conjecture*: If $\sum_{x \in A} \frac{1}{x}$ diverges then A has arbitrarily long arithmetic sequences. – 1
- What is the chromatic number of the plane? – 1 [This one is easy! The chromatic number of the plane is the least natural number c such that the plane can be c-colored with no two points an inch apart having the same color. So your problem is solved. Oh. You want to know what the least c actually is. Oh. That does sound hard!]

Other

All of these got one vote.

- Any problem whose answer is unclear.
- Does Psi Exist?
- Computer Vision.
- Self-driving cars.
- What goes on inside a computer and what can theoreticians do to make it better? (see my (Don Knuth's) keynote talk "Theory and Practice" from the 11th World Computer Congress, available as *Theoretical Computer Science*, vol. 90 (1991), pp. 1–15).

2.17 Anything Else You Want to Comment On?

This question has too many diverse answers to summarize them. I list some of them.

Dmytro Taranovsky:
Thanks for doing the survey. Without proofs, the best we have are the opinions of experts, including differences of opinion. [Especially the differences!]

Tom Schaefer:
A question you didn't ask: Will P=?NP *be resolved by a human, or by a machine operating autonomously? I lean toward the latter.*

Benoit Razet:
Thanks for running this survey. Can't wait to see the results! [Me too!]

Don Knuth:
I can't wait for people to realize that P=?NP *is only a question about the existence of a polynomial time algorithm, not about the knowledge of one. We already know, for example, that there exists a polynomial time algorithm to decide membership in any given minor-closed graph family; but we don't know how to really decide it, except for a few actual families.*

Jacob Kesinger:
For me the resolution of the question P=?NP *isn't particularly interesting. What I as an outsider to Theoretical Computer Science want is to see algorithmic improvements that make hard problems easy and practically-impossible problems feasible. More SAT solvers and fewer reductions to Coppersmith–Winograd matrix multiplication. [There are now matrix multiplication algorithms that do slightly better than Coppersmith–Winograd, but the point holds there as well – all of the algorithms are probably galactic.]*

Bogdan Grechuk:
I hope that all mathematicians enjoy the last decades in which mathematicians equipped with computer are significantly stronger in proving math theorems than computers alone. This will soon end. In 20 years, our role will be to press the button and either see the proof, or, if not, conclude that if a computer cannot prove the theorem then it is hopeless for humans to even start. Of course, there will be exceptions: today there are some special chess positions which humans understand

better than computers. In a similar way, tomorrow there will be some special kind of theorems which we can prove better. But this will be a rare situation.

Yuval Filmus:

The P=?NP *question is an interesting theoretical issue, but it is best to focus on understanding why SAT solvers work so well in practice. [Some of the comments on Question 15 indicate that SAT-solvers do not always do well in practice; however, Yuval's question, which I interpret as asking why SAT solvers work as well as they do, is very interesting.]*

Paul Beame:

We have now got a lot riding on SETH *(strong exponential time hypothesis). In some sense, these reductions suggest new SAT algorithms that people haven't often thought of previously. It seems plausible that* P≠NP *but* SETH *is false. It would be worth asking people about* SETH *(and* ETH). [If you remind me I'll ask that in my next poll.]

2.18 Some Questions for Statistical Purposes

The rest of the questions were for statistical purposes.

Do I have permission to print your response with your name? Without your name? Not at all?

93 responses

- With your name: 39 (42%)
- Without your name: 47 (50%)
- Not at all: 7 (8%)

What is your highest degree?

94 responses

- PhD: 61 (65%)
- Masters: 17 (18%)
- Bachelors: 15 (16%)
- I once had a fever of 100 degrees. In my delirium I though I proved P=NP: 1 (1%)

What is your degree in?

I suspect that many of the people who checked off just one of Computer Science or Mathematics actually majored in both.

100 responses

- Computer science: 64 (64%)

- Mathematics: 21 (21%)
- Both Computer Science and Mathematics: (5%)
- Physics: 4 (4%)
- Electrical Engineering: 3 (3%)
- Comp Bio: 1 (1%)
- Logic: 1 (1%)
- Economics: 1 (1%)
- Business: 1 (1%)

When was your highest degree awarded?

80 responses

- 0-1979: 3 (4%)
- 1980-1989: 12 (15%)
- 1990-1999: 9 (11%)
- 2000-2010: 20 (25%)
- 2011-2018: 34 (43%)

3 What do People Who Have Thought About P=?NP Think?

Some of the people I surveyed are known to have seriously thought about the problem. We call them *the experts*. I call the set of non-experts *the masses*. It is of interest to see how the opinions of the experts and the masses differed. I do this for some of the questions.

Does P=NP*?:*

1. The Experts: 99% NO, 1% YES.
2. The Masses: 87% NO, 13% YES.
3. Both groups were confident of their responses.

When will P=?NP *be resolved?:*

1. Experts: 55% before 2100, 45% after 2100.
2. Masses: 69% before 2100, 31% after 2100.
3. Neither group was confident of their responses.

Does P=BPP*?:*

1. Experts: 98% YES, 2% NO.
2. Masses: 60% YES, 40% NO.
3. The experts were confident of their responses. The masses were not.

Is Graph Isomorphism in P?

1. Experts: 70% YES, 23% NO, 7% DK. Gee, I don't know either, but I still answered. That's why it's called *an opinion poll*.
2. Masses: 40% YES, 60% NO.
3. Neither group was confident of their responses.

Is Factoring in P?

1. Experts: 31% YES, 62% NO, 7% DK.
2. Masses: 35% YES, 65% NO.
3. Neither group was confident of their responses.

4 Wrap Up

Well, that's it for now. If P=?NP is not resolved within the next 10 years I may do another survey. Or maybe a robot will do it for me.

5 Acknowledgment

I want to thank Hunter Monroe for setting up the Surveymonkey poll and for suggesting Question 12 on hard instances. I want to thank Clyde Kruskal for proofreading ω times, comments, and discussion. I want to thank David Sekora, Nathan Grammel, Justin Hontz, Karthik Abinav, Josh Twitty, and Lane A. Hemaspaandra. They proofread it sequentially (in the order of names given) after Clyde's ω-proofreading and yet still found some errors! And of course I want to thank Notorious LAH for giving me a forum to present my survey.