Open Problems Column
Edited by William Gasarch

1 This Issue’s Column!

This issue’s Open Problem Column is by William Gasarch and is My Answers to My P vs NP Poll.

2 Request for Columns!

I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere in between, and (2) really important or really unimportant or anywhere in between.

My Answers to My P vs NP Poll

By William Gasarch

3 Introduction

There have been two polls asking theorists (and others) what they thought of P vs NP (and other questions) [6, 7]. Both were written by William Gasarch (me) and appeared in the Complexity Column of SIGACT News, edited by Lane A. Hemaspaandra. They were in 2002 and 2012. Since William Gasarch is fond of Van der Waerden’s theorem, you would think the next poll would be 2022; however, Notorious LAN asked if it could be a bit earlier so, to coincide with his 100th column, and I agreed.

The results of that poll will appear in Notorius’s column in this issue. That article is written objectively – I report on what people responded without comments like Lance, really, there are intelligent people who think P = NP but do not think Elvis is alive. By contrast, this column will be my answers and perhaps some commentary on the answers others gave.

This column is about what I think the status of P vs NP is and will be. For information on what is the status of P vs NP see Scott Aaronson’s article [1], Lance Fortnow’s article [4], or (for the layperson) Lance Fortnow’s book [5].
4 Questions and Answers

1. Do you think P = NP? I think P \neq NP. I was once 200% sure now I am only 100% sure. The Graph Minor Theorem makes me ponder if some really hard math theorem my all of a sudden, and perhaps by accident, get SAT \in P. Also, people have been trying to get SAT into P for only about 50 years. A drop in the mathematical bucket? Or since they were 1970-2018 and not 1670-1718, perhaps more of a splash.

2. When do you think the question will be resolved?

I will first address who might solve it. Hilbert’s 10th problem was to find (in today’s terminology) an algorithm that will, given \( p \in \mathbb{Z}[x_1, \ldots, x_n] \) determine if \( p \) has a Diophantine solution. Hilbert likely never considered that there would be no such algorithm. Hence it took outsiders to think outside the box and lay the groundwork for the solution. In those days Martin Davis (a logician), Hilary Putnam (a philosopher), Julia Robinson (a logician and a women) were outsiders for the reasons given in parenthesis. So – do we need an outsider for P vs NP?

- YES- the current system rewards the-next-Stoc/focs-paper more than long term thought.
- NO- the math needed to work on H10 was not much. The math needed to work on P vs NP seems like it may be quite formidable. See next point.
- NO- the myth of the genius-in-his-basement is no longer true. For hard pure math was it ever true?

But there is a part 2 to the story of H10. David-Putnam-Robinson set the problem up, by Matiyasevich. And who was he? He was a 23-years old brilliant Russian Mathematician. More outside-the-box.

Okay, so when? H10 needed a few new ideas plus the genius of seeing that there was no such algorithm. But I keep coming back to the fact that it didn’t need that much math and P vs NP does. So when will it be solved:

- In the year 2525
  - If man is still alive
  - If women can survive
  - They may know that P \neq NP.
  - But we still won’t know the status of Graph Isomorphism.

3. (Answer this one only if you answered P \neq NP above.) Sasha Razborov, Avi Wigderson, and Andy Yao (or three other wise people whose opinions on P vs NP you take seriously) all knocked at your door at 3 a.m. to tell you that P vs NP has been resolved — but after announcing it dashed 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 \neq NP or your believe that we are no where near proving P \neq NP.) In the actual poll more people thought P \neq NP then I would have thought. This may be because the three wise men all think P \neq NP. If I ask this question again on my next poll I may make it a group with people on both sides of the issue.

While I believe P \neq NP, I have a stronger believe that we are nowhere near a proof. Hence I would think P = NP. Therefore I would cancel all of my credit cards and go back to sleep.
4. What kind of mathematics will be used to resolve P vs NP? Let us separate out what I want to see happen vs what I think will happen. I hope the answer depends on Ramsey Theory and Computability theory since that way I might be able to understand it. In my dreams the ideas come out of a RATLOCC (Ramsey Theory, Logic, and Complexity) meeting. Okay, now back to reality.

It has been said that combinatorics is both the easiest and hardest field of mathematics. Easy since a lot of it requires no prerequisite knowledge. Hence a High School Student can do work in it. Hardest because a lot of requires no prerequisite knowledge. Hence you can’t easily apply continuous techniques. As a concrete example, \( \sum_{i=1}^{n} i^{100} \) is hard, whereas \( \int_{1}^{100} x^{100} \, dx \) is easy. Having said that, continuous techniques have been used more and more in combinatorics. I believe they will continue to do so. So I’ll go with some combination of combinatorics and continuous math. That probably covers all of mathematics.

Another thought: Perhaps showing SAT \( \not\in P \) is hard but showing that Factoring cannot be done in polynomial time is easier. It may be that Factoring is the key problem, not SAT.

5. Do you think the polynomial hierarchy collapses to some level (e.g., there is an \( i \) such that \( \Sigma^p_i = \Sigma^p_{i+1} \))? I think \( \Sigma^p_i \neq \Sigma^p_{i+1} \) and it will be proven the same time as \( P \neq NP \). Why are they different? Because of all those thousands of problems that are \( \Sigma^p_5 \)-complete that mathematicians (even before Cook’s Theorem) and Computer scientists have been trying to get into \( \Sigma^p_4 \). Oh. There aren’t any? Hmmm. Maybe I’m not that confident after all.

6. Do you think that SAT has polynomial-sized circuits?

I think SAT does not have polynomial-sized circuits. That’s my advice on advice. So why not? I can imagine advice helping factoring (e.g., a table of certain kinds of numbers). I just can’t imagine advice helping with SAT.

I am more confident that SAT does not have poly-sized circuits than I am that PH does not collapse. Hence the Karp-Lipton Theorem:

\[
SAT \in P/poly \rightarrow \Sigma^p_i = \Pi^p_i
\]

always looked odd to me. It says something really unlikely implies something less unlikely.

7. Do you think P = BPP?

YES. This is one of those rare statements that the community changed its mind on. In 1985 most people thought \( P \neq BPP \) ( alas I do not have a poll to prove it, but I was in the community at the time). Mike Sipser was an exception [9]. After the Nisan-Wigderson Hard implies Random results [8] the community shifted to thinking \( P = BPP \). I can imagine this one being proved within the next 5 years; however, I said that 10 years ago.

In this years poll the trend seems to be reversing- 60% thing \( P \neq BPP \). This surprised me.

8. Do you think that SAT \( \in BQP \) (commonly called Quantum P) implies that the polynomial hierarchy collapses? I hope so, if only so when students claim they read on the web that
SAT is in Quantum P. I can tell them with more authority NO. Not that many problems are in Quantum P. (When I’ve blogged and said this people attack me and tell me to go back to blogging about the Pumping Lemma. When Scott blogs and says that people treat the statement and him with the respect it deserves.) I doubt SAT is one of them. Its even sort-of an accident that Factoring is in Quantum P.

9. Do you think P = NP \cap \text{coNP}? No. Since I first cut my teeth on computability theory I was hoping the answer is yes. But since factoring is in P = NP \cap \text{coNP}, and I don’t think factoring is in P. I’ll have to say NO. Oh well.

10. Do you think Graph Isomorphism is in P? No; however, I do think it is in P \cap \text{coNP} by derandomization. I’ve heard that Babai’s result [2] that GI is in quasipolynomial time \( n^{(\log)^O(1)} \) is as far as current methods can go. Hence it will be a long time before a new advance. Or not.

As noted above, I hope that even after P vs NP is resolved the status of GI is unknown. If so will people lose interest?

Worst Case Scenario: GI \in P and the proof uses the classification of finite simple groups. If so this might be the largest constant ever in an algorithm.

11. Do you think factoring is in polynomial time? No. This is probably the question where I am least confident of my answer. Contrast the following:

- A logician uses logic to show SAT \in P. I rather doubt this will happen. I doubt that SAT is even a problem in logic. In fact, deep methods in logic have not given us any faster algorithms for SAT.
- A number theorists uses number theory to show factoring is in polynomial time. This is quite plausible. In fact, deep methods in number theory have given us faster algorithms for factoring.

12. If someone shows P = NP will this have a big effect on practical computing?

ANSWER: YES. While the first algorithm for SAT \in P may well be terrible, the ideas behind it will be used to get practical algorithms

13. If you answered P \neq NP above do you believe that an obstacle is “hard instances,” for example, for any deterministic Turing machine M accepting the language

\[ L = \{(N,x,1^t): \text{Nondeterministic } N \text{ does not halt on input } x \text{ within } t \text{ steps }\} \]

there exists \( (N',x') \) such that the runtime of M on \( (N',x',1^t) \) is not bounded by a polynomial \( t^c \)?

Yes. While the first algorithm for SAT \in P may well be terrible, the ideas behind it will be used to get practical algorithms for SAT and other problems.

One chapter of Lance Fortnow’s book [5] on P vs NP is a fictional story of what happens after P = NP is proven. Initially the algorithm is terrible. But since P = NP they use the
algorithm to find a better one. After many iterations they have a really fast algorithm but have no idea why it works. While I doubt that will happen, something similar might.

Don Knuth, Clyde Kruskal, and others think that even if P = NP there will be a notion of hardness-within-P, and the problems that are now NP-complete will still be hard. This is plausible.

14. If someone shows P ≠ NP will this have a big effect on practical computing?

Maybe. Sorry for the cop-out answer. The techniques may give us such great insights into computation that they are used to solve other problems. Or not.

I’ve had the following conversation with my darling

BILL: If P ≠ NP the proof can’t help but have such great insights that will help real world programmers write better and faster code.

DARLING: I think it can help it

She may be right.

15. Given that SAT-SOLVERS are now quite good, will P vs NP become less relevant?

Yes. SAT-SOLVERS are great on some problems; however, my own experience with trying to use them on Ramsey Theory problems indicates that they have their limits.

The premise that they are “now quite good” was challenged by some of the people who answered the poll. Factoring can be phrased as a problem in SAT, and they haven’t done well there. More generally there may be a street-light -problem here. Recall the anecdote:

Alice to Bob: Why are you looking for your keys under the street light when you dropped them a block away?

Bob to Alice: The light is better here.

16. Aside from P vs NP which open problem do you most want to see solved?

(a) The Erdos-Turan Conjecture: show that if \( \sum_{x \in A} \frac{1}{x} \) diverges then A has arbitrarily long arithmetic progressions.

(b) The Erdos-Turan-Gasarch Conjecture: show that if \( \sum_{x \in A} \frac{1}{x} \) diverges then, for all \( p_1, \ldots, p_k \in \mathbb{Z}[x] \) with \( p_i(0) = 0 \), there exists \( a, d \) such that \( a, a + p_1(d), \ldots, a + p_k(d) \in A \).

(c) Obtain tight asymptotic bounds on the Van Der Warden Numbers and the polynomial VDW numbers.

(d) Do the last item with a proof I can understand!

(e) Prove that computing the Ramsey Numbers is hard. This may require a new framework for complexity.

(f) Resolve the difficulty of The Muffin Problem (see [3]).

17. Anything else you want to comment on, feel free!

I wish that the popular media did a better job at portraying the problem. All mentions of it in fiction have been terrible. One problem is that we are trying to prove that problems are hard which is not as sexy as proving problems are easy.
18. Do I have permission to print your response with your name? without your name? not at all? *Duh*

19. What is your highest degree? What is it in? Where is it from? When did you get it? The answer will not appear in the article; however, I want it for statistical use.

*PhD from Harvard in Computer Science in 1985.*

References


