The Art of Writing Technical Blog Posts

Parth Parikh

Microsoft - Summer ‘25 Intern Talk

Why should you write blog posts?

Clarity of Thought

Writers v Readers

TABLE 1. HOW MANY PEOPLE HAVE EVER LIVED ON EARTH?

YEAR	POPULATION	BIRTHS PER 1,000	BIRTHS BETWEEN BENCHMARKS	NUMB
190,000 B.C.E	2	80	–	–
50,000 B.C.E.	2,000,000	80	7,856,100,000	7,856,1
8000 B.C.E.	5,000,000	80	1,137,789,769	8,993,889,771
1 C.E.	300,000,000	80	46,025,332,354	55,019,222,125
1200	450,000,000	60	26,591,343,000	81,610,565,125
1650	500,000,000	60	12,782,002,453	94,392,567,578
1750	795,000,000	50	3,171,931,513	97,564,499,091
1850	1,265,000,000	40	4,046,240,009	101,610,739,100
1900	1,656,000,000	40	2,900,237,856	104,510,976,956
1950	2,499,000,000	31-38	3,390,198,215	107,901,175,171
2000	6,149,000,000	22	6,064,994,884	113,966,170,055
2010	6,986,000,000	20	1,364,003,405	115,330,173,460
2022	7,963,500,000	17	1,690,275,115	117,020,448,575

Population Reference Bureau Data

Newsletters

The Atlantic

TECHNOLOGY

Google: There Are Exactly 129,864,880 Books in the World

By Alexis C. Madrigal

AUGUST 5, 2010

SHARE SAVE

If 30% of the people born since 1850 each published three pieces of writing:

(2900237856 + 3390198215 + 6064994884 + 1364003405 + 1690275115) * 0.3 * 3 * 1500 / 90000

~ 231M books
or ~ 7 Library of Congress'

Why do I write blog posts?

Title	Published
The Secret World of Data Structures and Algorithms 📌	22/06/23
My Link Log 📌	09/11/19
What if companies were governed as republics?	26/05/25
An Afternoon with SMT, TRIM, and SMART	23/02/25
Tinker Postman Collector Scribe	13/12/24
We'll buy back your Typewriter ... for Uncle Sam!	02/10/24
In the land of LLMs, can we do better mock data generation? ↗	27/09/24
IBM 305 RAMAC and the 1960 Winter Olympics	03/09/24
One Billion Row Challenge in Rust	23/04/24
Computing the Hausdorff Distance **pdf**	20/03/24
Roaring Bitmaps	28/02/24
Tangoing with a Martin Gardner Word Game	28/01/24
Scrambling Eggs for Spotify with Knuth's Fibonacci Hashing	08/12/23
Bao, a learned system for query optimization	17/11/23
Before advanced-flight simulators, there was Navitrainer	18/09/23
Before Xerox, there was Addressograph	25/06/23
GADDAG	22/06/23
Playing Twenty Questions with ChatGPT	17/03/23
What do Kirchhoff, Cayley, and Carroll have in common? And who is Kahrimanian?	26/02/23
Odd Sketches	19/02/23
An architecture for enforcing RBAC in a cloud storage system	17/02/23
KHyperLogLog	24/01/23

Manuel Blum's Advice to Graduate Students

You are all computer scientists.
You know what FINITE AUTOMATA can do.
You know what TURING MACHINES can do.
For example, Finite Automata can add but not multiply.
Turing Machines can compute any computable function.
Turing machines are incredibly more powerful than Finite Automata.
Yet the only difference between a FA and a TM is that
the TM, unlike the FA, has paper and pencil.
Think about it.
It tells you something about the power of writing.
Without writing, you are reduced to a finite automaton.
With writing you have the extraordinary power of a Turing machine.

https://www.cs.cmu.edu/~mblum/research/pdf/grad.html

This applies to reading as well

Consider writing what you read as you read it.
This is especially true if you're intent on reading something hard.

I remember a professor of Mathematics at MIT,
name of BERTRAM KOSTANT,
who would keep his door open whenever he was in his office, and he would always be at his desk writing.
Writing. Always writing.
Was he writing up his research? Maybe.
Writing up his ideas? Maybe.
I personally think he was reading, and writing what he was reading.
At least for me, writing what I read is one of the most enjoyable and profitable ways to learn hard material.

https://www.cs.cmu.edu/~mblum/research/pdf/grad.html

California gold rush: Research papers meet Real world

Eh.., What's Technical, Doc?

CS166
Spring 2021

Handout 09
April 13, 2021

Suggested Project Topics

Here is a list of data structures and families of data structures we think you might find interesting topics for your research project. You're by no means limited to what's contained here; if you have another data structure you'd like to explore, feel free to do so!

My Wish List

Below is a list of topics where, each quarter, I secretly think "I hope someone wants to pick this topic this quarter!" These are data structures I've always wanted to learn a bit more about or that I think would be particularly fun to do a deep dive into.

You are not in any way, shape, or form required to pick something from this list, and we aren't offering extra credit or anything like that if you do choose to select one of these topics. However, if any of them seem interesting to you, we'd be excited to see what you come up with over the quarter.

- Bentley-Saxe dynamization (turning static data structures into dynamic data structures)
- B^ε-trees (a B-tree variant designed to minimize writes)
- Chazelle and Guibas's $O(\log n + k)$ 3D range search (fast range searches in 3D)
- Crazy good chocolate pop tarts (deamortizing binary search trees)
- Durocher's RMQ structure (fast RMQ without the Method of Four Russians)
- Dynamic prefix sum lower bounds (proving lower bounds on dynamic prefix parity)
- Farach's suffix tree algorithm (a brilliant, beautiful divide-and-conquer algorithm)
- Geometric greedy trees (lower bounds on BSTs giving rise to a specific BST)
- Ham sandwich trees (fast searches in 2D space, not along axis-aligned boxes)
- Iacono's unified structure (fast searches for items near items looked up recently)
- Kaplan and Tarjan's fast catenable deques
- Multisplay trees (nesting splay trees inside one another)
- Odd sketches (estimate Jaccard similarity through hash collision parities)
- Round-elimination lower bounds (what limits are there on sorted dictionaries?)
- Sequence heaps (priority queues optimized for the memory hierarchy)
- Strict Fibonacci heaps (achieving Fibonacci heap bounds, worst-case)
- Succinct RMQ (solving RMQ efficiently with as few bits as possible)
- Tabulation hashing (a simple hashing technique more powerful than it might initially seem)
- Top trees (data structures for storing information about changing graphs)
- Wavelet trees (storing trees in as few bits as possible)
- Weak AVL trees (modern balanced BST that might replace red/black trees)

Approximate Distance Oracles Stories from Knuth's Books

KHyperLogLog RBAC Odd Sketches

SQL query optimization GADDAG

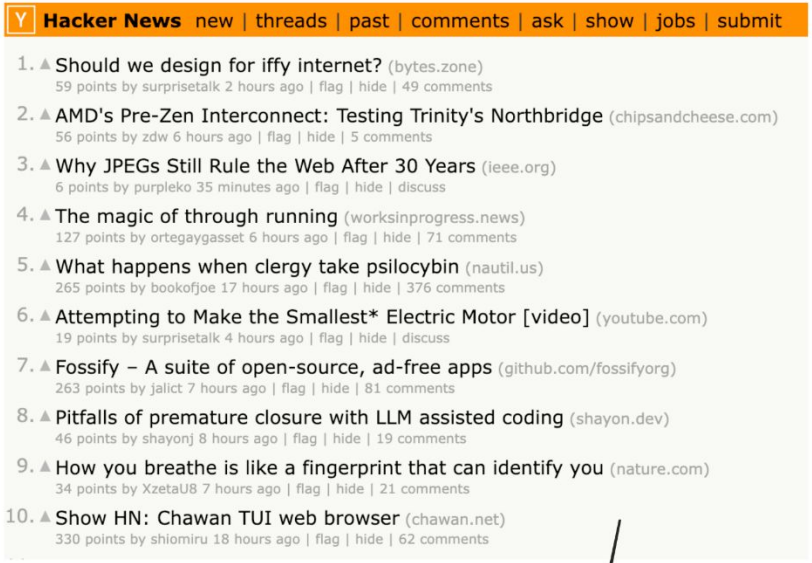
Knuth's Fibonacci Hashing Roaring Bitmaps Navitrainer

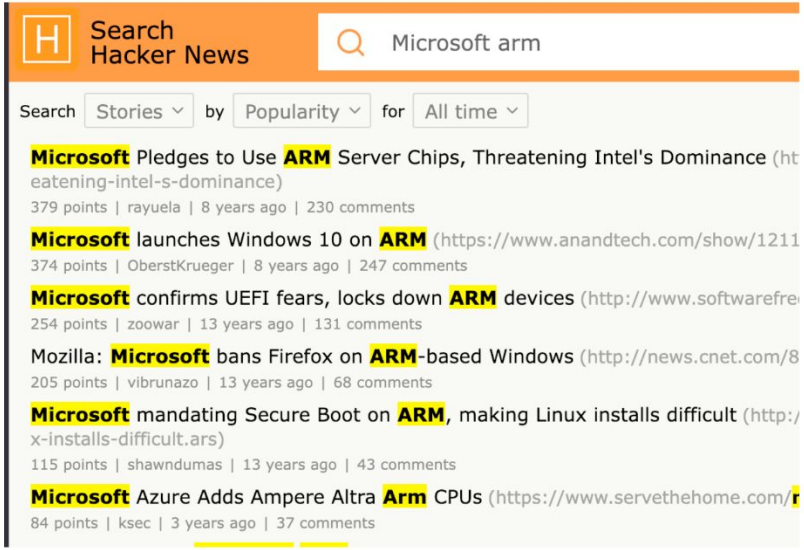
Can be for MBA topics, Economic Think Tanks, you name it!

https://web.stanford.edu/class/archive/cs/cs166/cs166.1216/handouts/090%20Suggested%20Project%20Topics.pdf

Show me the source!

Diverse sources == Diverse perspectives in your writing





Not for citation, for perspective

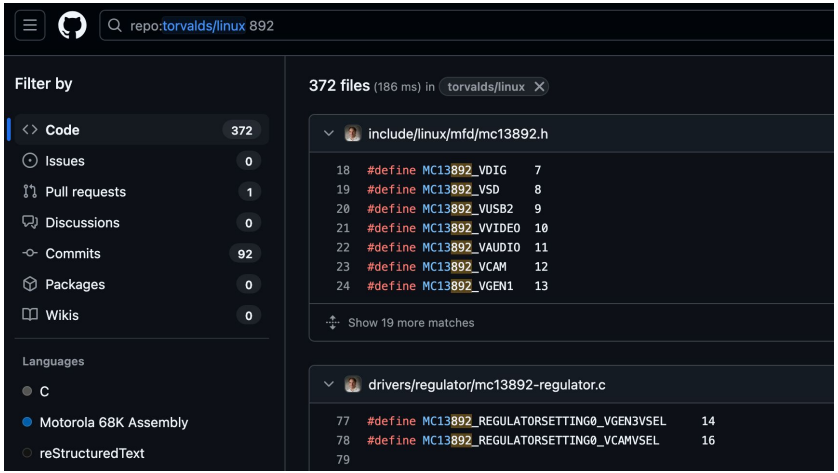
Ask: Where will I find high signal to noise ratio?

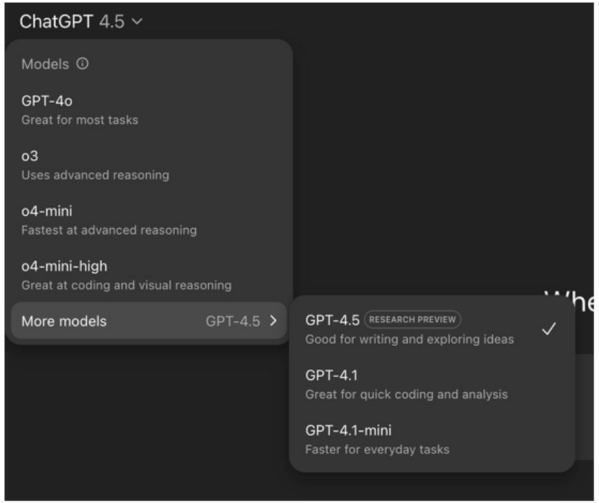
More sources?

Google Scholar

Articles Case law

Stand on the shoulders of giants





LocalLLaMA

leave 487,828 readers

721 users here now

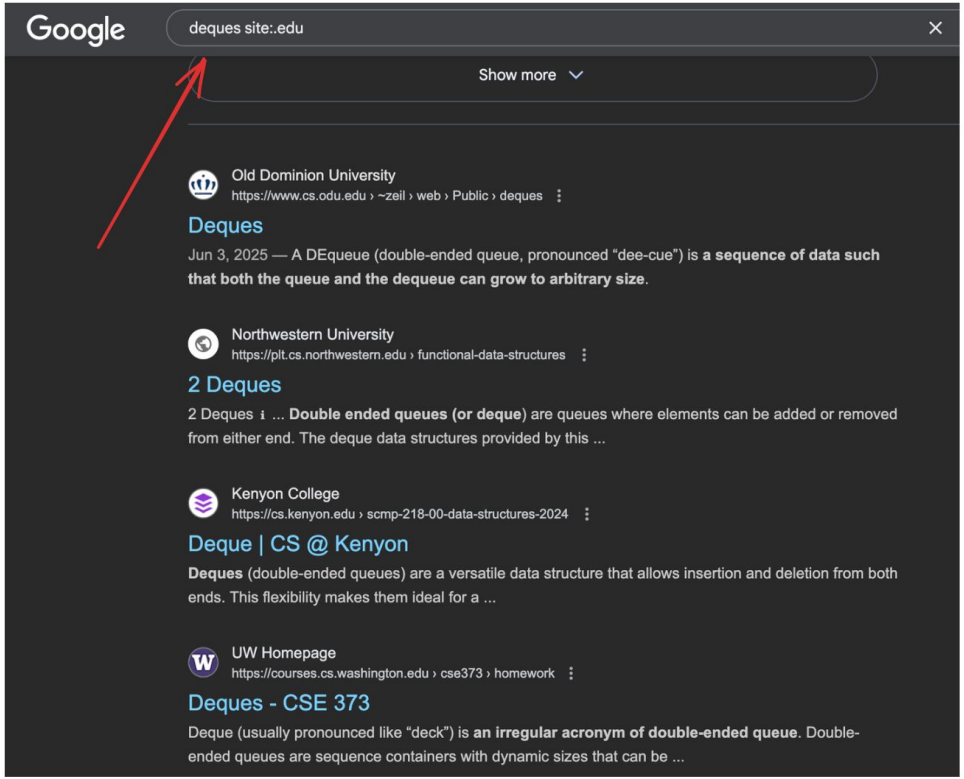
Show my flair on this subreddit. It looks like: pncnmnp (edit)

r/LocalLLaMA

A subreddit to discuss about Llama, the family of large language models created by Meta AI.

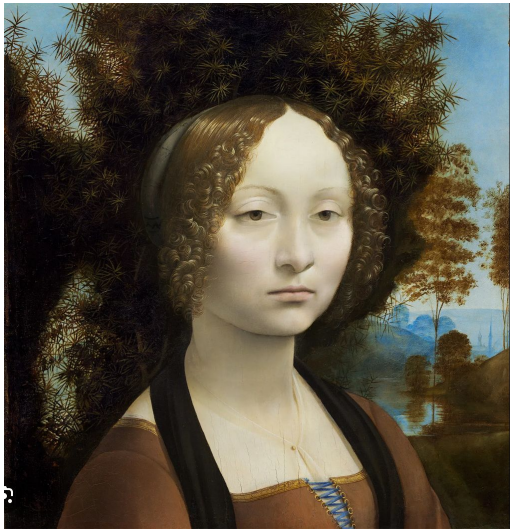
Drill Down!

Be specific with Search



Ahem, Parth? How do I write? Show me how to write?

Write however you want



My learnings from blogging

- Keeping the blogging stack minimal helps.
  - Using simple tools has helped me focus entirely on the content. For instance, I write everything in Google Docs and then manually convert it to HTML.
- It is beyond okay to feel stuck, especially with technical content.
  - I often have several partially written drafts sitting around. Revisiting these drafts periodically helps me see them with fresh perspectives. Eventually, inspiration strikes, and I end up finishing those half-baked drafts.
- It helps to avoid obsessing over analytics.
  - I have intentionally avoided analytics, and it has kinda allowed me to focus on topics that genuinely interest me, rather than writing solely to please some imaginary audience. It kind of gives me the freedom to explore obscure subjects.

Questions?