

the adversary game

part II:

Bits of Information

last time...

Shall we play a game?

twenty questions...

I guess a number N between 1 and 2^{20} .
you ask yes/no questions of the form "is $N \leq \text{blah}$ ",
and after 20 questions you must tell me what N is.



Hmm, I see what he did there.
He did not really fix a single "true" N...

He answered so that
the "set of valid N's" was at least
half of the previous set.

Since at least two valid Ns remained, I lost.



this Claude kid's sharp!

OK, ok, enough with these childish games, young Claude.

How about we play a real game?

tell me...



I think of integer N in $[1, 2^{\{20\}} + 1]$...

But now you can ask me any twenty
yes/no questions about N .

After 20 questions you must tell me what N is.

Should we play?



what do you think?

Nah, you're just messing with me.
Even with these rich questions, I need 21 steps.

how would you argue that?

See, you started with $M = 1 + 2^{20}$ options for N
For each question, you can choose the yes/no answer that
leaves at least $\text{ceiling}(N/2)$ options still valid.

After 20 questions, at least $\text{ceiling}(M/2^{20}) = 2$ options
will be still valid, and I will lose.



*valid = satisfy all the questions asked so far.

To be able to figure out which of M options you're thinking of using only yes/no questions.

I need $\text{ceiling}(\log_2 M)$ questions.

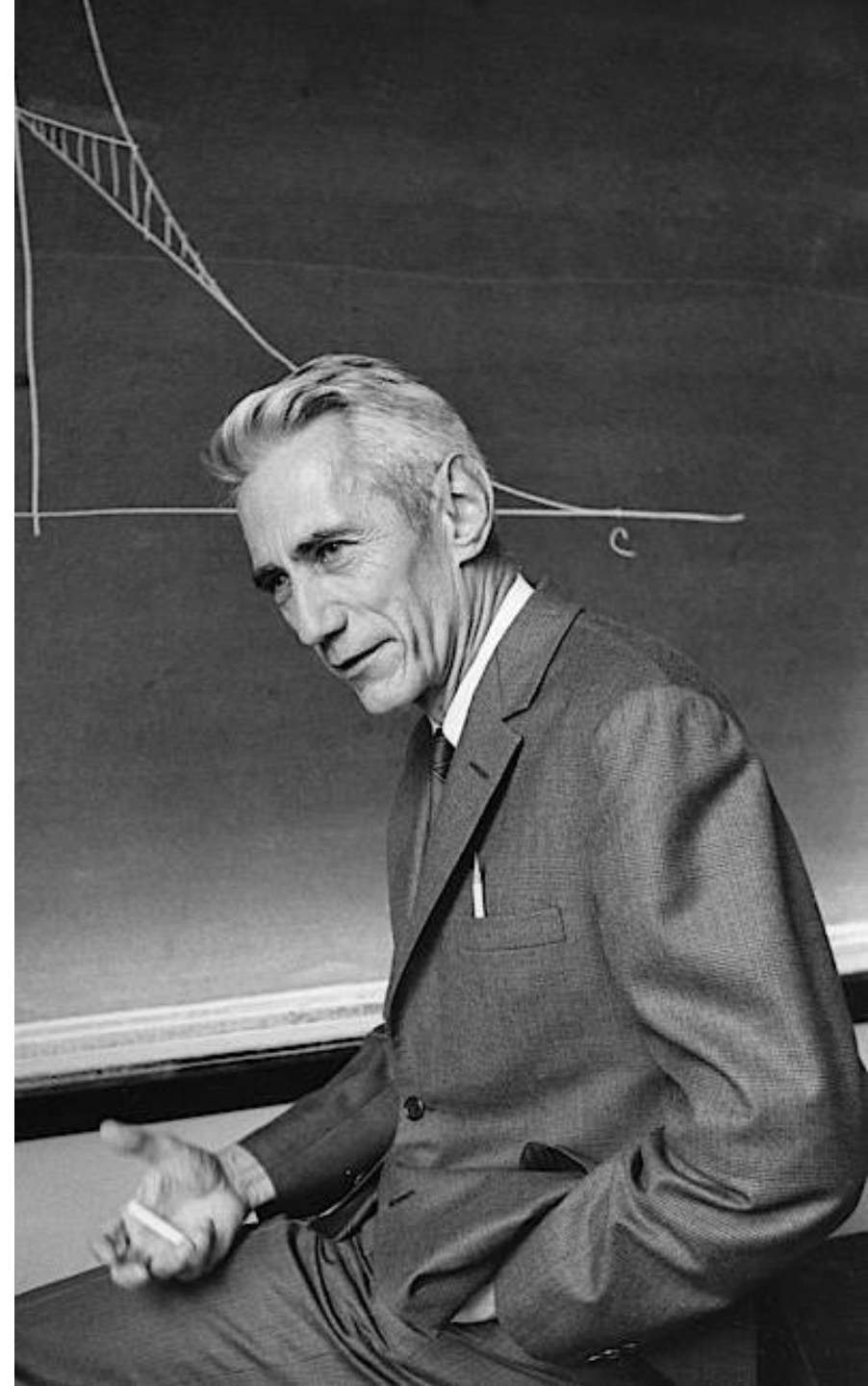
If we're trying to sort n elements, I am trying to figure out which of the $M = (n!)$ possible inputs have been given.

A comparison-based algorithm learns about the input via (a special kind of) yes/no answers.

So I need at least $\text{ceil}(\log_2 (n!))$ questions in this model.

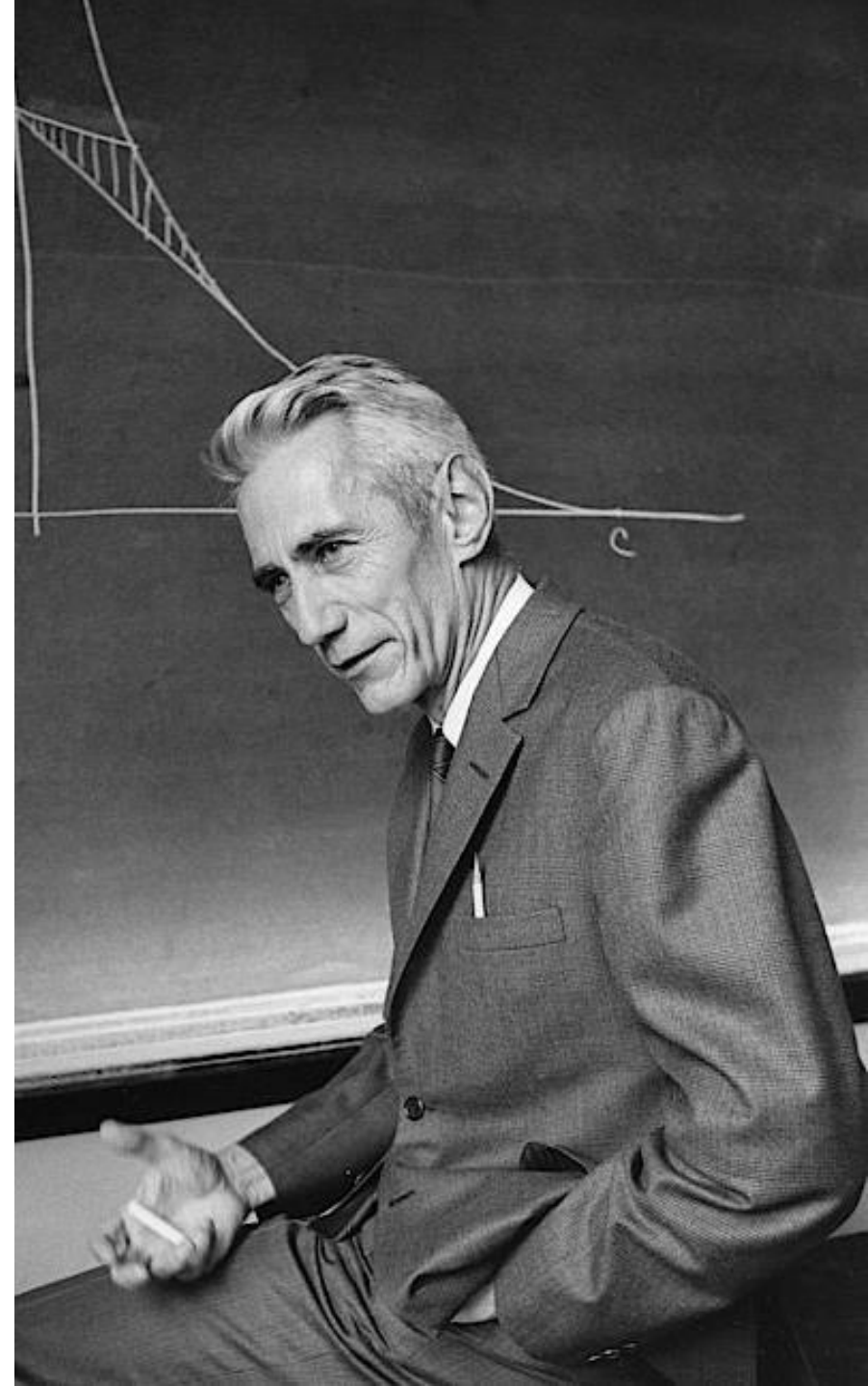


this is called the "information theoretic" bound
(and Claude is C. Shannon, who invented info.theory)



here's a different "CS-y" way to think about the bound.

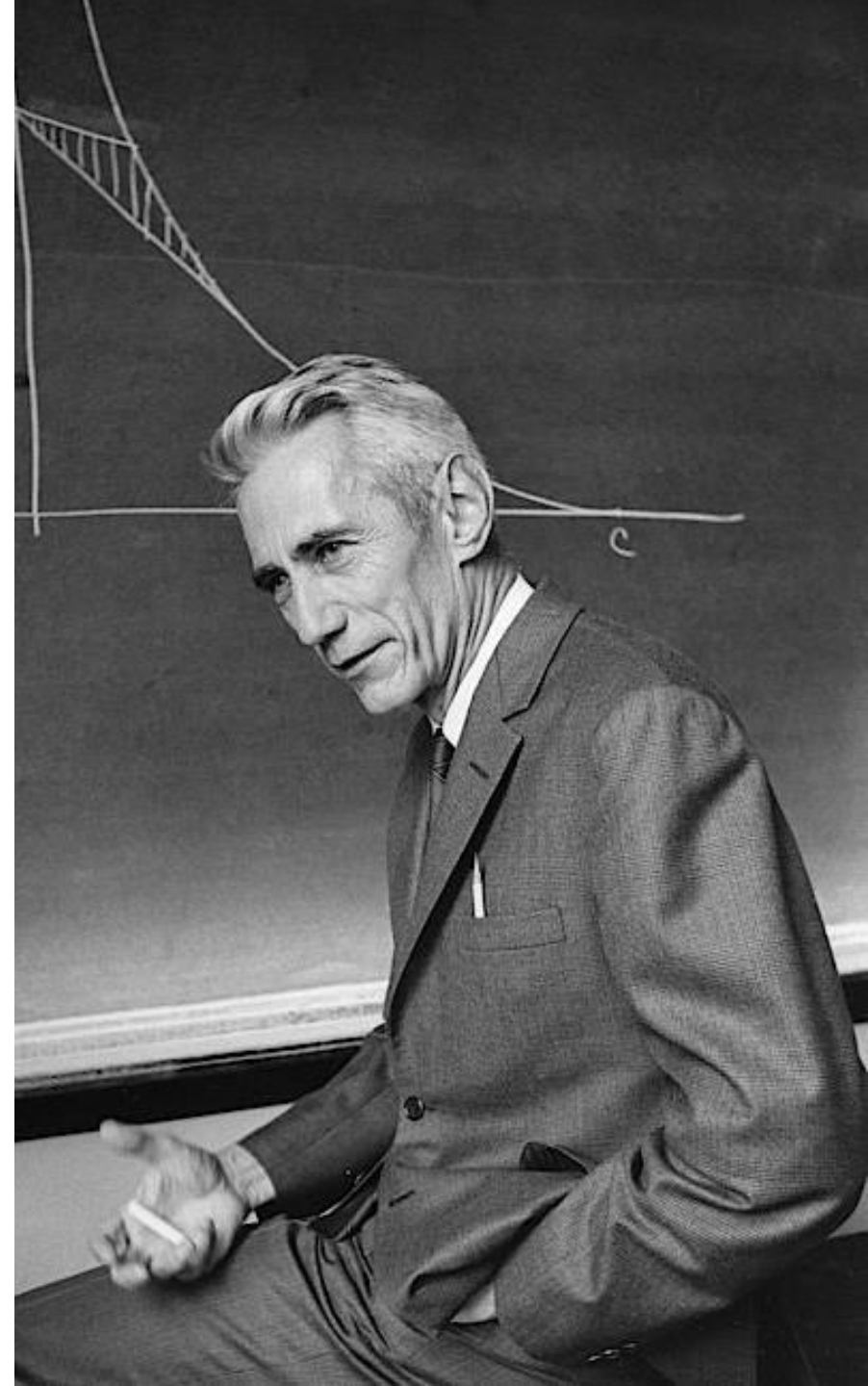
1. you're trying to guess the name of one of M items.
2. to write down M distinct bit strings needs $\geq \text{ceil}(\log_2 M)$ bits
3. each yes/no query gives you at most 1 bit of information
4. so you need at least $\text{ceil}(\log_2 M)$ queries for some item.



And this bound is super-general

The objects you are trying to guess may be
permutations (in sorting)
or the location of the max (in max-finding)
or whatever you want.

If there are M of them, and you ask binary questions
you need $\text{ceil}(\log_2 M)$ questions to pin one down.



OK, smartypants, why does this fancy info.theory of yours not give tight bounds for finding the maximum in the comparison model?

Simply because it's so general.

It says that no matter what yes/no questions I ask, finding the maximum will take $\text{ceil}(\log_2 n)$ time.



And I can ask you $\text{ceil}(\log_2 n)$ yes/no questions to find the max.

(I can ask: is the max in the first half?
If so, is it in the first quarter, etc.)

Just these queries are complex,
they are not comparisons between elements.

And I cannot always halve the options each time using comps,
so you can make me take longer.



You're a worthy pupil!
Let me teach you something else.

How to show that max-finding takes $n-1$ comp. queries
or max-and-min finding takes $(3/2)n - 2$ comps.

using "my" argument...



Oooh, I've always wanted to learn the adversary argument...



what is the "adversary" argument?

will they use a "potential function"?

will Claude be lured to the dark side?

stay tuned for the answers...



featuring

BSD "beastie" deamon	...	the adversary
Claude Elwood Shannon	...	Claude

BSD Daemon image from [Wikipedia](#) (which is from the CD-ROM for FreeBSD 2.0)
based on original artwork by Phil Foglio

Shannon images from the Shannon Centennial Celebration [webpage](#) at U.Mich, and the MIT Museum
xkcd font by Randall Monroe, at <https://github.com/ipython/xkcd-font>