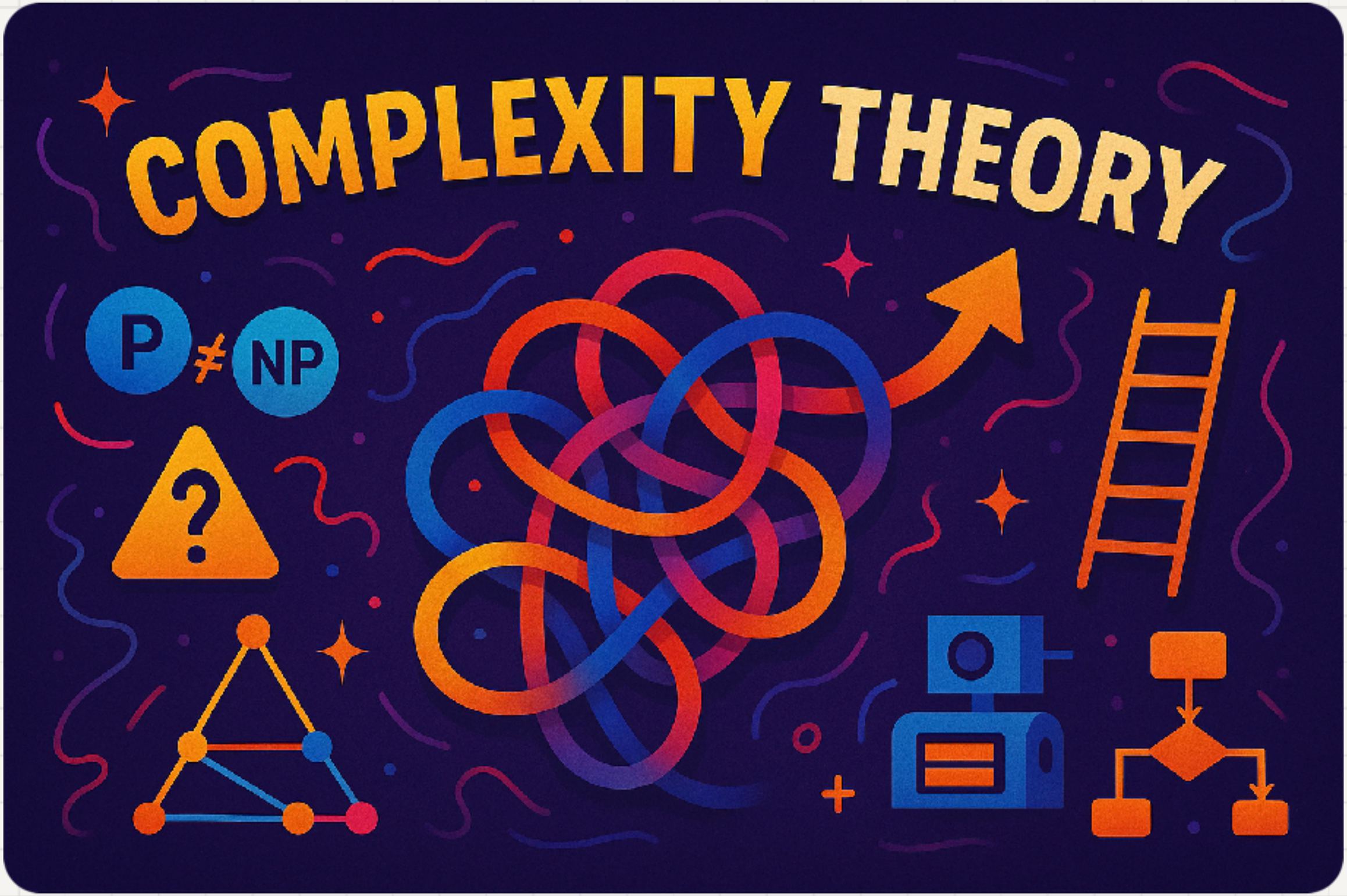


Introduction

Tom Gur

The story begins here in Cambridge

From Computation Theory to Complexity Theory

Turing, 1936

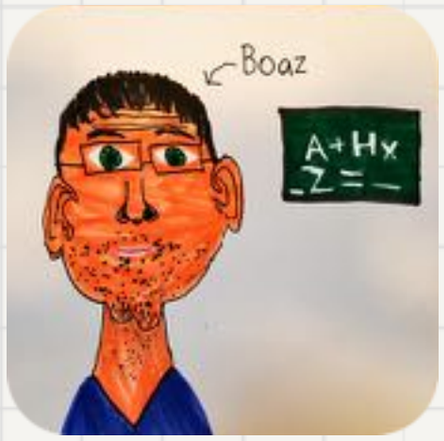
Finite or infinite, that is the question.

Hartmanis and Stearns, 1965

Or is it?

Numbers and scale

Ryan O'Donnell
CMU



Boaz Barak
Harvard

10 - Fingers

100 - a full lecture theatre

1000 = 2^10 - a rock concert

1000000 = 2^20 - 5x population of Cambridge, 1M seconds = 11.5 days

1 Billion = 2^30 - 1B seconds = 31.5 years, 1GHz = 1/billion sec, iPhone

1 Trillion = 2^40 - 1T seconds = 34,842.1 years, FLOPS of PlayStation

2^50 = 35,678,377.2 years, FLOPs of a supercomputer

2^60 - 36 Billion years, 3 times the age of the universe

2^150 - all super computers working throughout the universe's lifespan...

2^1000000 complexity of an exponential-time algorithm on a small input...

What is Complexity Theory

Complexity theory characterises tractable computation!

Why care?

Practice: Learn how to recognise intractable problems.

Theory: Understanding the power and limitations of computation,

inspiring new paradigms

Interdisciplinary: Deep connections to physics and mathematics

Bonus: Ways to make progress on big philosophical questions:

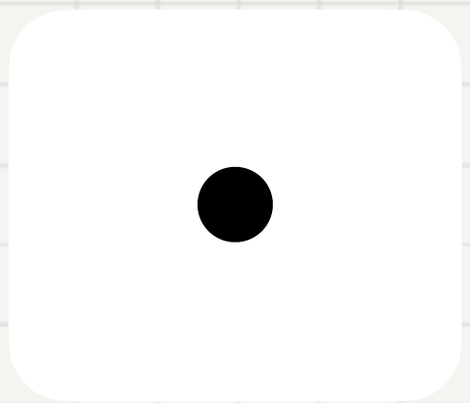
randomness, quantum, free will, and beyond

Scott Aaronson
UT Austin

The power of abstraction

Avi Wigderson
Princeton

Without enough abstraction, it is hard to make progress



With excessive abstraction, we lose complexity

Asymptotics

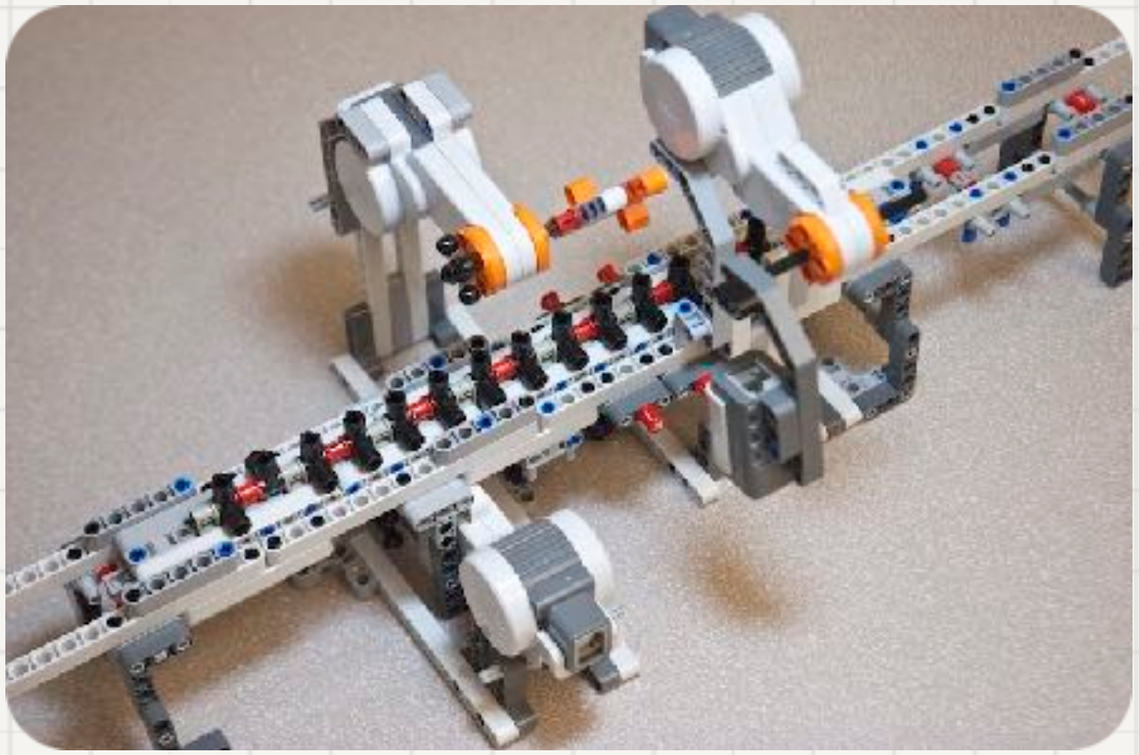
Definition

For functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N} \to \mathbb{N}$, we say that:

- $f = O(g)$, if there is an $n_0 \in \mathbb{N}$ and a constant c such that for all $n > n_0$, $f(n) \leq cg(n)$;
- $f = \Omega(g)$, if there is an $n_0 \in \mathbb{N}$ and a constant c such that for all $n > n_0$, $f(n) \geq cg(n)$.
- $f = \theta(g)$ if $f = O(g)$ and $f = \Omega(g)$.

The power of abstraction

Turing Machines



- Q — a finite set of states;
- Σ — a finite set of symbols, including $\sqcup$ and $\triangleright$.
- $s \in Q$ — an initial state;
- $\delta : (Q \times \Sigma) \to (Q \cup \{acc, rej\}) \times \Sigma \times \{L, R, S\}$

$$(q, w, u) \to_M (q', w', u')$$

if

- $w = va$;
- $\delta(q, a) = (q', b, D)$; and
- either $D = L$ and $w' = v$ and $u' = bu$
  or $D = S$ and $w' = vb$ and $u' = u$
  or $D = R$ and $w' = vbc$ and $u' = x$, where $u = cx$. If u is empty, then $w' = vb\sqcup$ and u' is empty.

The language $L(M) \subseteq \Sigma^*$ accepted by the machine M is the set of strings

$$\{x \mid (s, \triangleright, x) \to_M^* (acc, w, u) \text{ for some } w \text{ and } u\}$$

Why TM?

Church-Turing thesis: a function can be computed iff it is TM-computable

Extended Church-Turing thesis: adds efficiency to the picture

(Quantum Computing is the only widely accepted contender to ECT)

Turing fact of the day

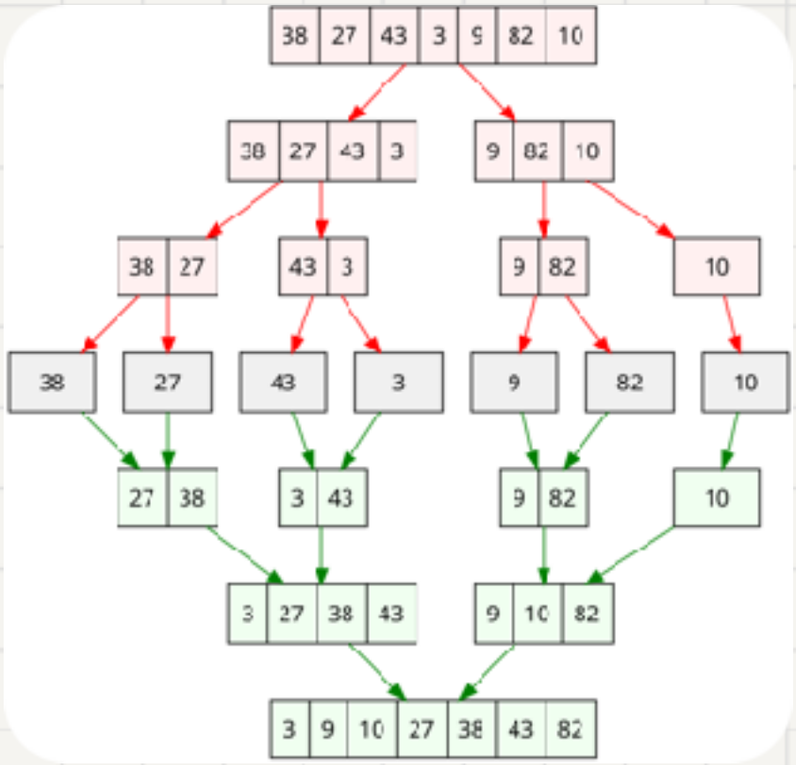
Ran a marathon in 2:46:03.

Considered representing England in the 1948 London Olympics.

Algorithms & Problems

Bubble Sort runs in time O(n^2)

Merge Sort in O(n log n)



We compare the two algorithms as they solve the same problem.

But what is the complexity of the sorting problem?

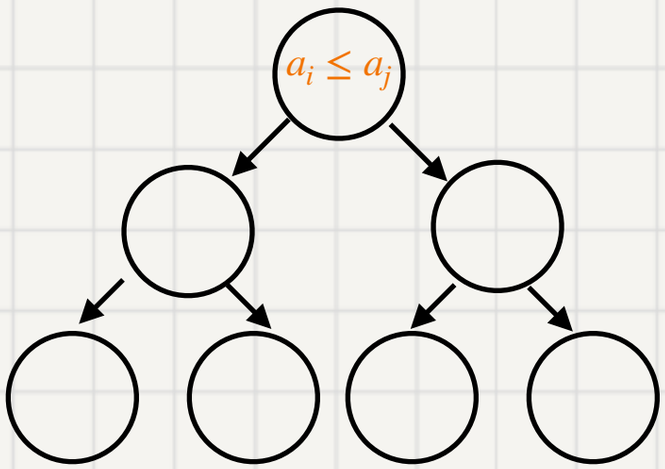
Upper bounds & lower bounds

What is the time complexity of the sorting problem?

Merge Sort gives us an upper bound of $O(n \log n)$

Showing a lower bound requires arguing about all algorithms

Further abstraction: Decision Trees



A comparison algorithm sorts n numbers

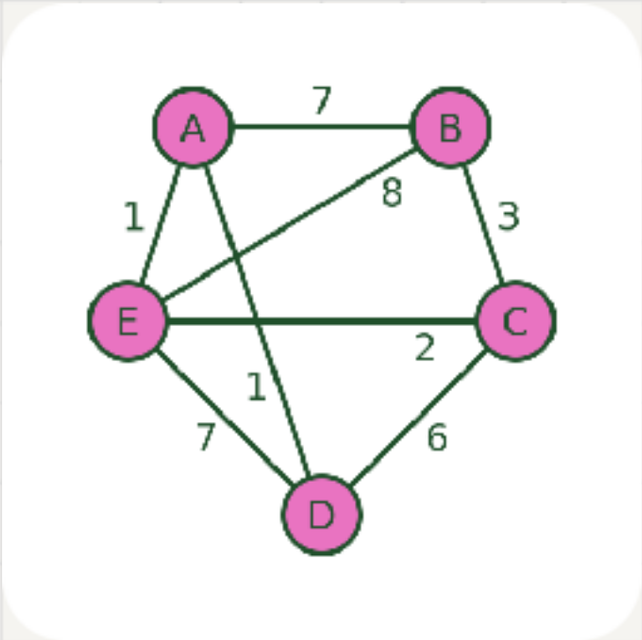
$a_1, \ldots, a_n$ by comparing pairs

There are $n!$ permutations, hence

tree height $\geq \log(n!) = \Omega(n \log n)$

Bonus: break the abstraction!

Travelling Salesman Problem (TSP)



Given a weighted graph with vertices $v_1, \ldots, v_n$,

Is there an ordering $\sigma \in S_n$ s.t.

$$\sum_{i=1}^{n} w\big(\sigma(i), \sigma(i+1)\big) \leq \tau$$

Complexity of brute force: $O(n!)$

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

Best lower bound*: $\Omega(n)$

Between these two is the chasm of our ignorance

About the course

Textbooks

"Computational Complexity: A Modern Approach" by Arora and Barak

"Computational Complexity: A Conceptual Perspective" by Oded Goldreich

Exam

Everything covered in the lectures

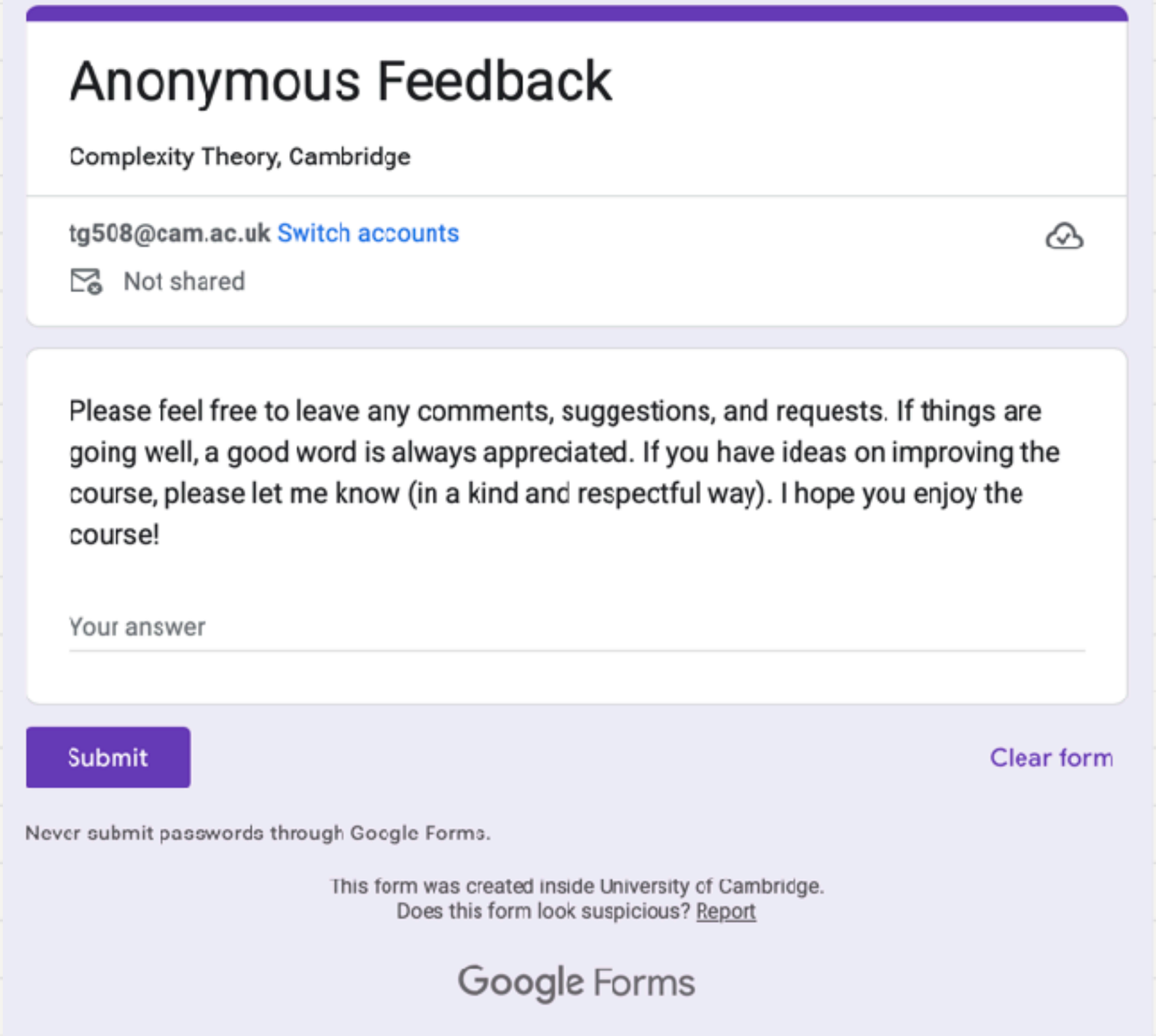
How to ace Complexity Theory

Internalising ideas vs problem solving

Participating in class

Active learning

Anonymous feedback



Anonymous Feedback

Complexity Theory, Cambridge

tg508@cam.ac.uk Switch accounts

Not shared

Please feel free to leave any comments, suggestions, and requests. If things are going well, a good word is always appreciated. If you have ideas on improving the course, please let me know (in a kind and respectful way). I hope you enjoy the course!

Your answer

Submit

Clear form

Never submit passwords through Google Forms.

This form was created inside University of Cambridge.
Does this form look suspicious? Report

Google Forms

https://www.cl.cam.ac.uk/teaching/2425/Complexity/materials.html