Mathematics & Computer Science: The World Through an Algorithmic Lens

Lev Reyzin

UIC

The Algorithmic Lens

- The Mathematics and Computer Science (MCS) major combines the two disciplines into an exciting program here at UIC.

- A main goal of our MCS major is to teach our students to “think computationally.”

What is Computational Thinking?

- Are there any problems we can't ever solve?
- Can two strangers share secrets in the open?
- If I proved the Riemann hypothesis, could I convince my colleagues without revealing anything about the proof?
- If white can always win in chess, can an omnipotent being quickly prove it?

What is Computational Thinking?

- **Are there any problems we can't ever solve?**
- Can two strangers share secrets in the open?
- If I proved the Riemann hypothesis, could I convince my colleagues without revealing anything about the proof?
- If white can always win in chess, can an omnipotent being quickly prove it?

Hilbert (1928)

Hilbert's Decision Problem

Entscheidungsproblem (decision problem)

"The decision problem is solved if one knows a process which, given a logical expression, permits the determination of its validity… we want to make it clear that for the solution of the decision problem a process would be given by which nonderivability can, in principle, be determined, even though the difficulties of the process would make practical use illusory… the decision problem [is] designated as the main problem of mathematical logic."

Richard Feynman (1918-1988)

Albert Einstein Award (1954)
E. O. Lawrence Award (1962)
Nobel Prize in Physics (1965)
Oersted Medal (1972)
National Medal of Science (1979)

Richard Feynman (1918-1988)

Albert Einstein Award (1954)
E. O. Lawrence Award (1962)
Nobel Prize in Physics (1965)
Oersted Medal (1972)
National Medal of Science (1979)

Feynman's Process

1) Write down the problem.
2) Think real hard.
3) Write down the solution.

(according to Gell-Mann)

Hilbert's ~~Decision~~ Problem

Entscheidungsproblem (decision problem)

"*The decision problem is solved if one knows a process which, given a logical expression, permits the determination of its validity… we want to make it clear that for the solution of the decision problem a process would be given by which nonderivability can, in principle, be determined, even though the difficulties of the process would make practical use illusory… the decision problem [is] designated as the main problem of mathematical logic.*"

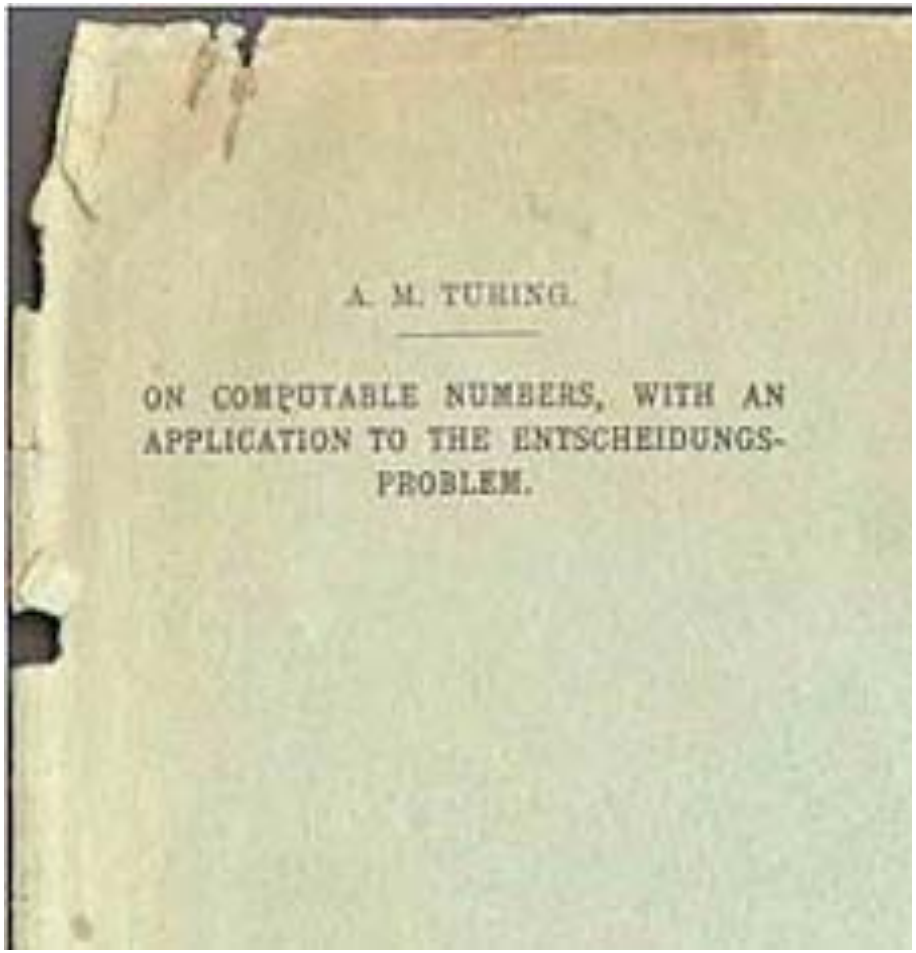
Alan M. Turing (1936)

Alan M. Turing (1936)

Church-Turing thesis: anything that is "effectively computable" is computable by this machine.

Alan M. Turing (1936)

Can just think about our favorite programming language, or any (even natural) process; it's all the same!



This is what allows for the algorithmic lens!

Turing's Answer to Hilbert

Hilbert's "process" does not exist for all problems!

Turing's Answer

Hilbert's "process" does not exist for all problems!

Some programs Halt:

```
program_1(x) {
  print("hello");
  for (i=0 to x){
    print(i);
  }
  exit();
}
```

Some programs don't:

```
program_2(x) {
  while(x<x+1) {
    print("nom ");
  }
  print("goodbye");
  exit();
}
```

The Halting Problem

Given a program P, how can we tell if it halts on x?

```
P(x) {
   int y = 40;
   while(x > y || y < 100) {
      for(int z = 1; z < x+y; x++) {
         y = y + 2;
         …
      }
      …
   }
}
```

The Halting Problem

Imagine there exists H(P,x) that given a program P and input x, returns 1 if P(x) halts and 0 if P(x) loops forever.

if such an H exists, we can create these two programs:

```
stops_on_self(P){
  return H(P,P);
}
```

```
garblygook(P) {
  if (stops_on_self(P))
    while(true) {};
  else
    exit();
}
```

The Halting Problem

Imagine there exists H(P,x) that given a program P and input x, returns 1 if P(x) halts and 0 if P(x) loops forever.

if such an H exists, we can create these two programs:

```
stops_on_self(P){
  return H(P,P);
}
```

```
garblygook(P) {
  if (stops_on_self(P))
    while(true) {};
  else
    exit();
}
```

What happens when you run `garblygook(garblygook)`?

The Halting Problem

Imagine there exists H(P,x) that given a program P and input x, returns 1 if P(x) halts and 0 if P(x) loops forever.

if such an H exists, we can create these two programs:

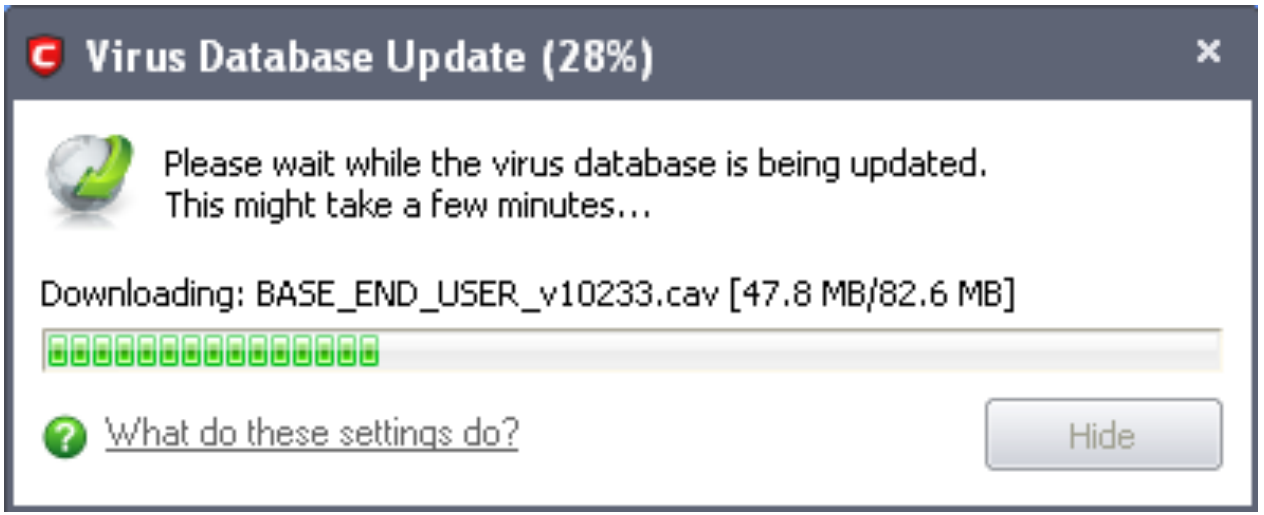
```
stops_on_self(P){
  return H(P,P);
}
```

```
garblygook(P) {
  if (stops_on_self(P))
    while(true) {};
  else
    exit();
}
```

What happens when you run `garblygook(garblygook)`?

So What?

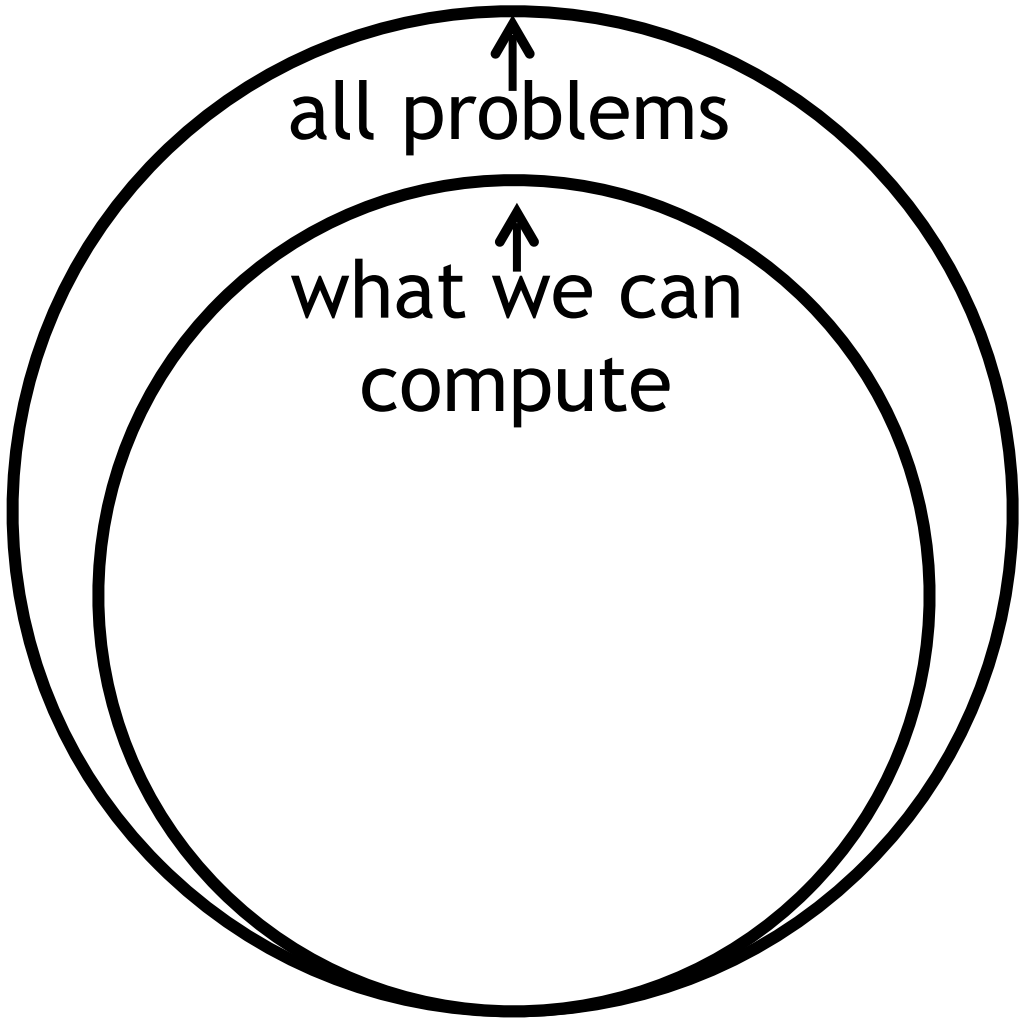
Tons of real-life applications: e.g. computer viruses.



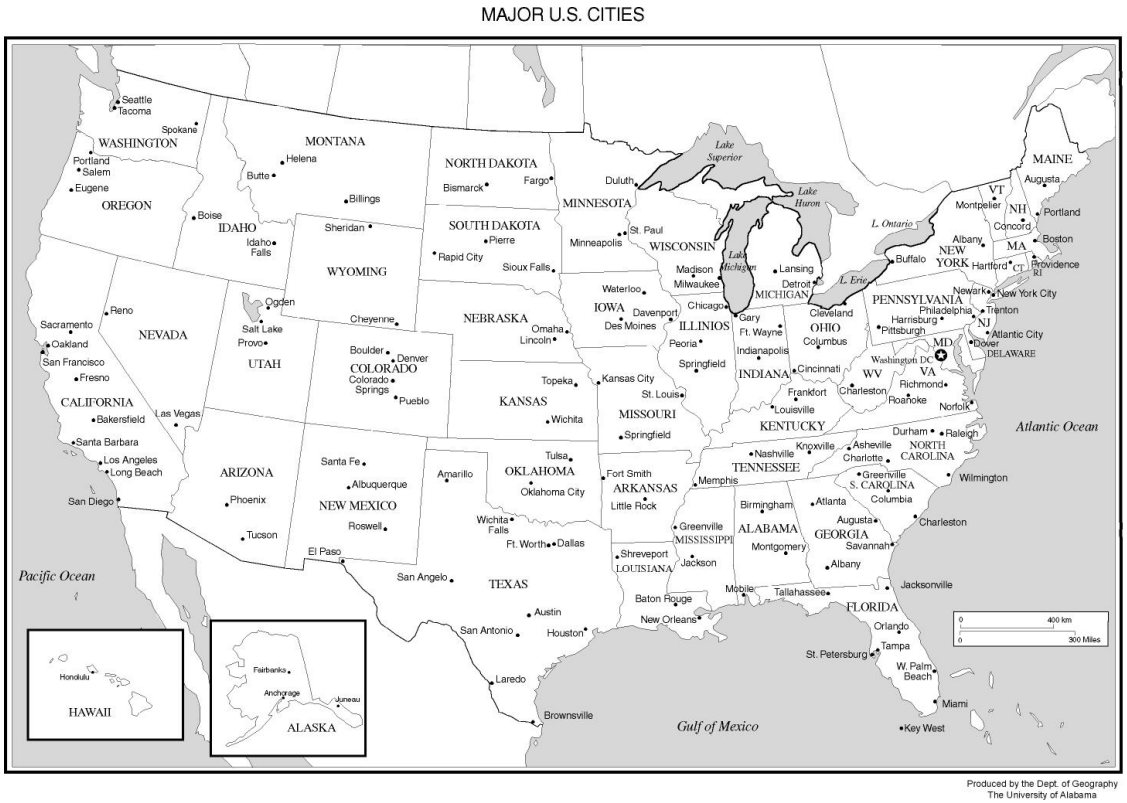
What is Computational Thinking?

- Are there any problems we can't ever solve?
- **Can two strangers share secrets in the open?**
- If I proved the Riemann hypothesis, could I convince my colleagues without revealing anything about the proof?
- If white can always win in chess, can an omnipotent being quickly prove it?

Computability Theory

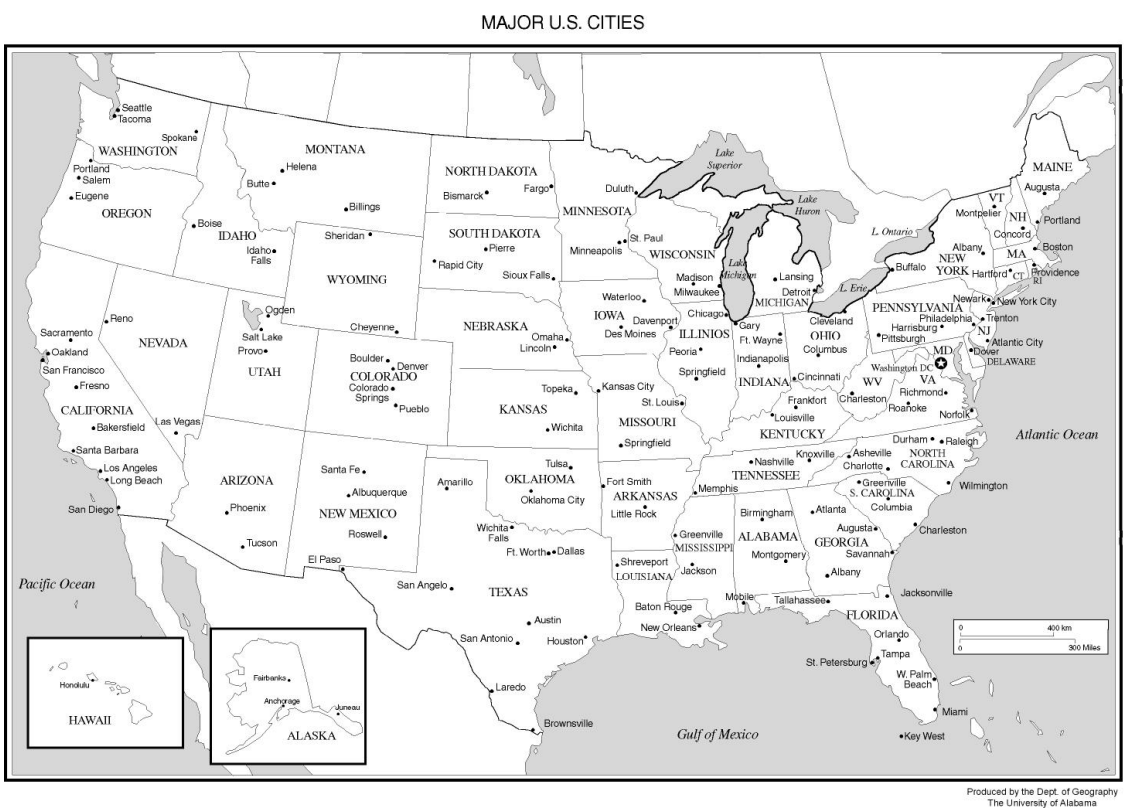


What is the Shortest Route Visiting all Major US Cities?



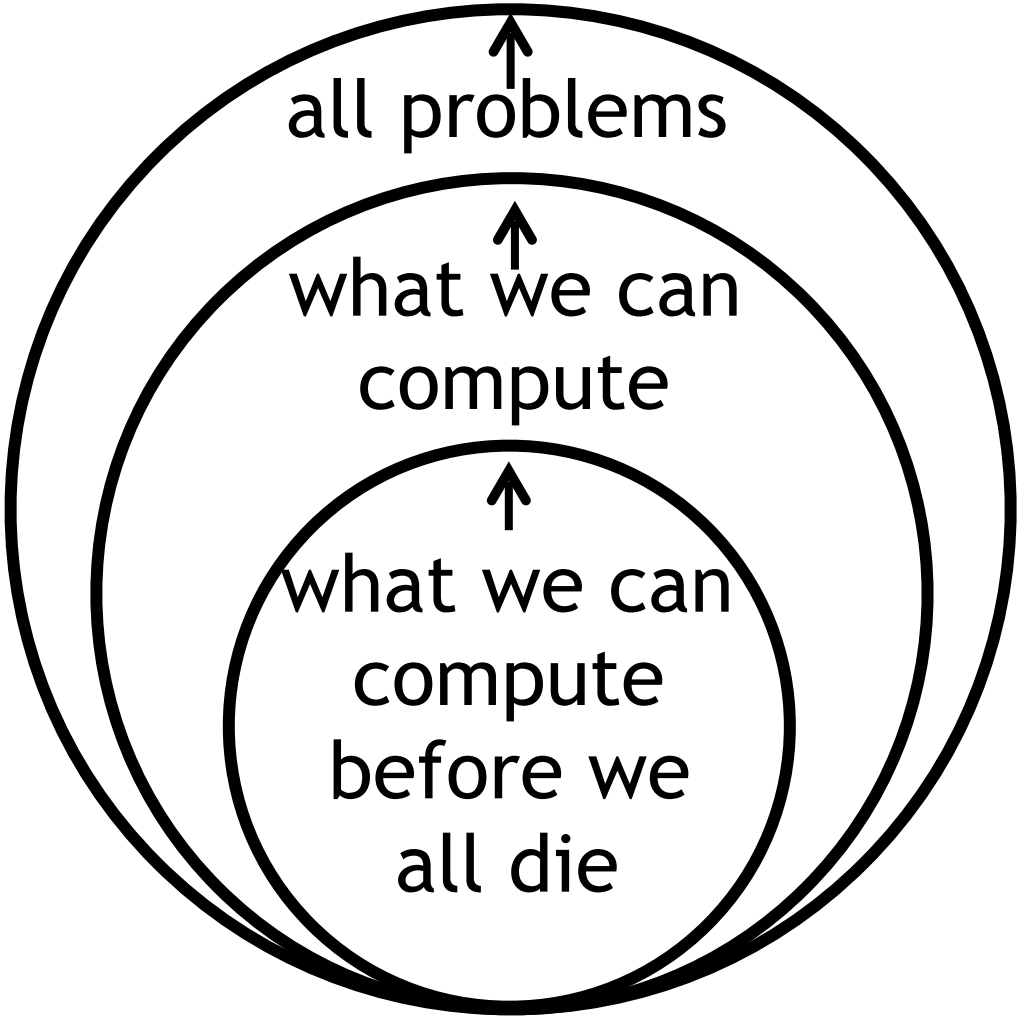
Can we try all routes and take the best?

What is the Shortest Route Visiting all Major US Cities?

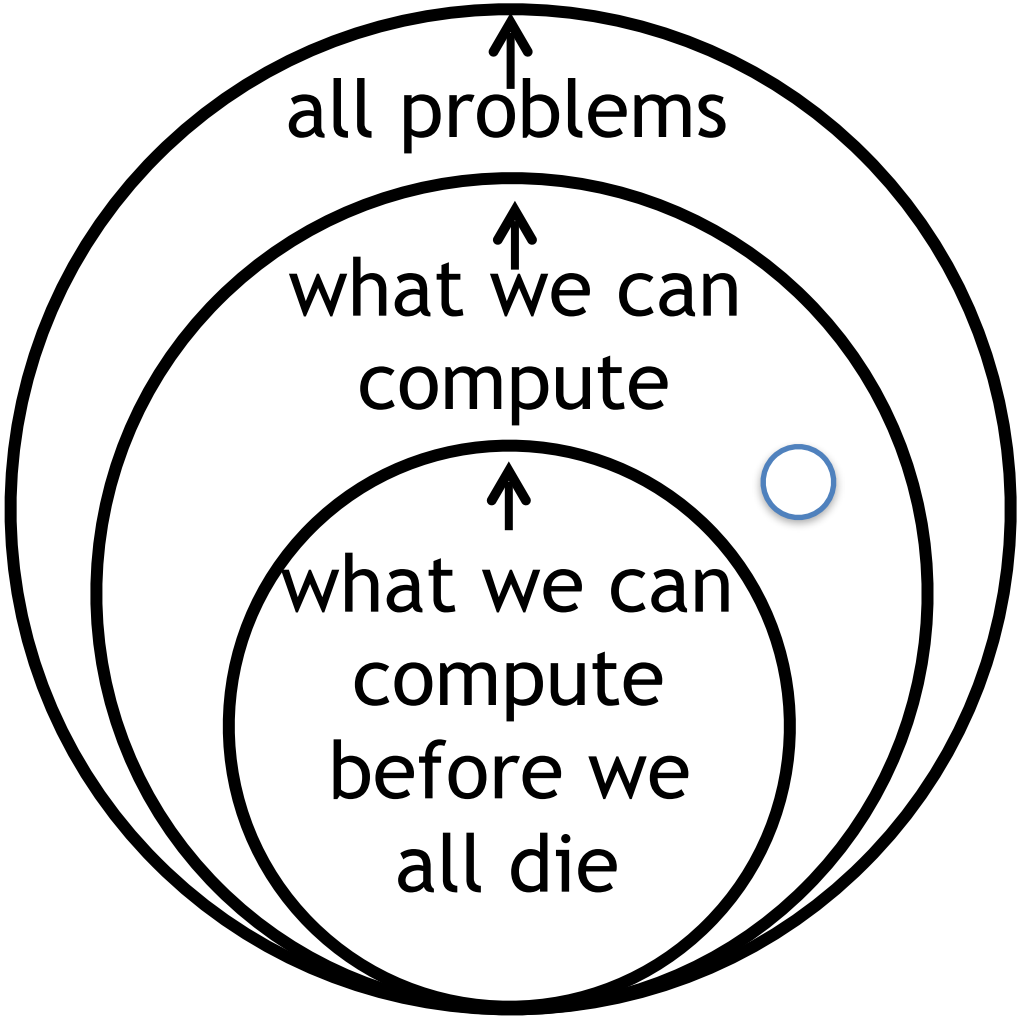


For 60 cities this would be 60! > 10^{81}, which happens to be 10 times the estimated number of atoms in the universe.

Complexity Theory



Complexity Theory



Using Hardness

Imagine Alice, Bob, and Eve all walk into a room. They have never met before. Everything they say to each other is heard by all 3. Can Alice and Bob exchange a secret that Eve doesn't know?

Alice

Eve

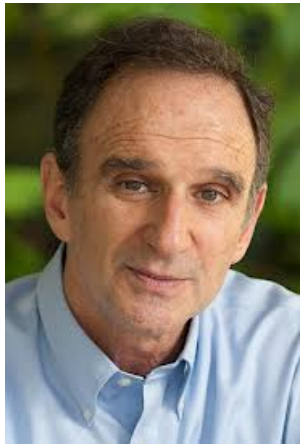
Bob

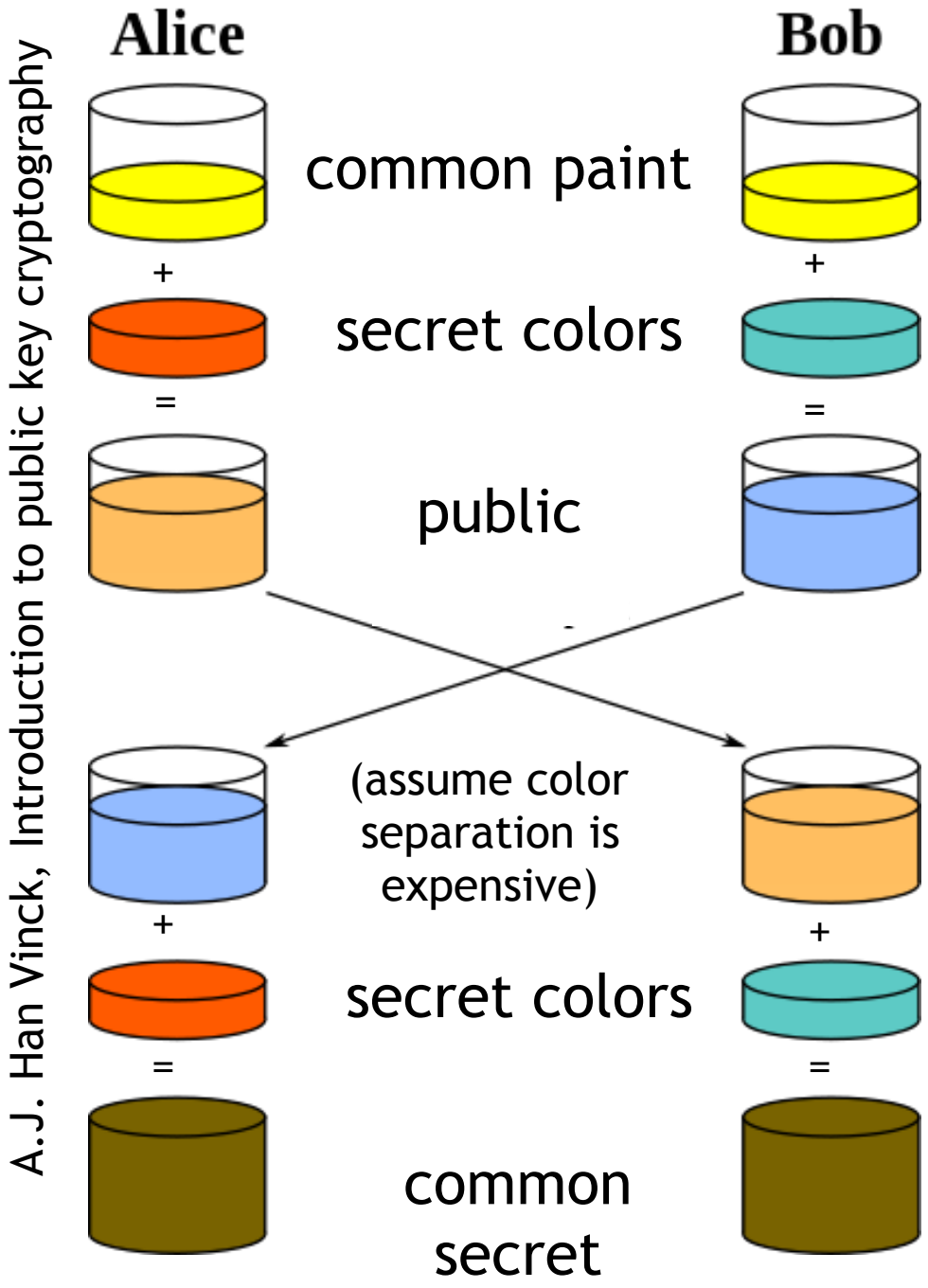
Using Hardness

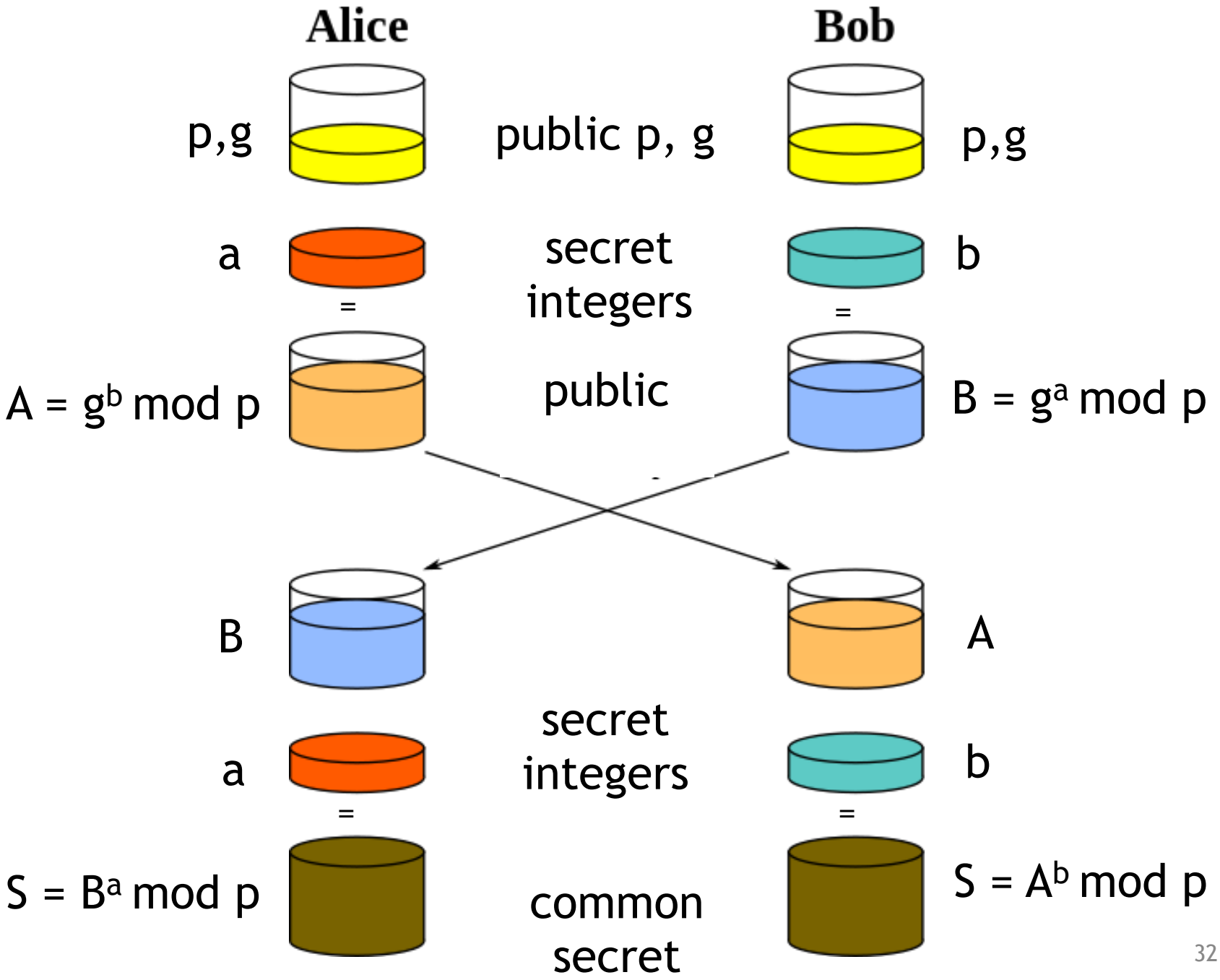
Imagine Alice, Bob, and Eve all walk into a room. They have never met before. Everything they say to each other is heard by all 3. Can Alice and Bob exchange a secret that Eve doesn't know?

Shockingly, Yes!

[Diffie-Hellman '76]




A.J. Han Vinck, Introduction to public key cryptography
Alice
Bob
common paint
+
secret colors
=
public
-
(assume color separation is expensive)
+
secret colors
=
common secret


Alice
Bob
p,g
public p, g
p,g
a
secret integers
b
=
=
A = g^b mod p
public
B = g^a mod p
B
A
a
secret integers
b
=
=
S = B^a mod p
common secret
S = A^b mod p

Using Hardness

Diffie-Hellman assumption (variant):

given
an integer g
a prime p,
g^x mod p and g^y mod p,
finding g^{xy} mod p cannot
be done efficiently

If the Diffie-Hellman assumption true,
Eve cannot know the common secret without
doing an exponential amount of computation.

Using the Shared Secret

Alice and Bob can then use their shared secret value for symmetric cryptography.

CAST5

AES

RC4

Blowfish

Serpent

IDEA

3DES

Twofish

What is Computational Thinking?

- Are there any problems we can't ever solve?
- Can two strangers share secrets in the open?
- **If I proved the Riemann hypothesis, could I convince my colleagues without revealing anything about the proof?**
- If white can always win in chess, can an omnipotent being quickly prove it?

Zero Knowledge Proofs

Let’s say Alice proves the Riemann Hypothesis. Can she convince Bob her proof is correct without revealing anything about her proof?

Zero Knowledge Proofs

Let's say Alice proves the Riemann Hypothesis. Can she convince Bob her proof is correct without revealing anything about her proof?

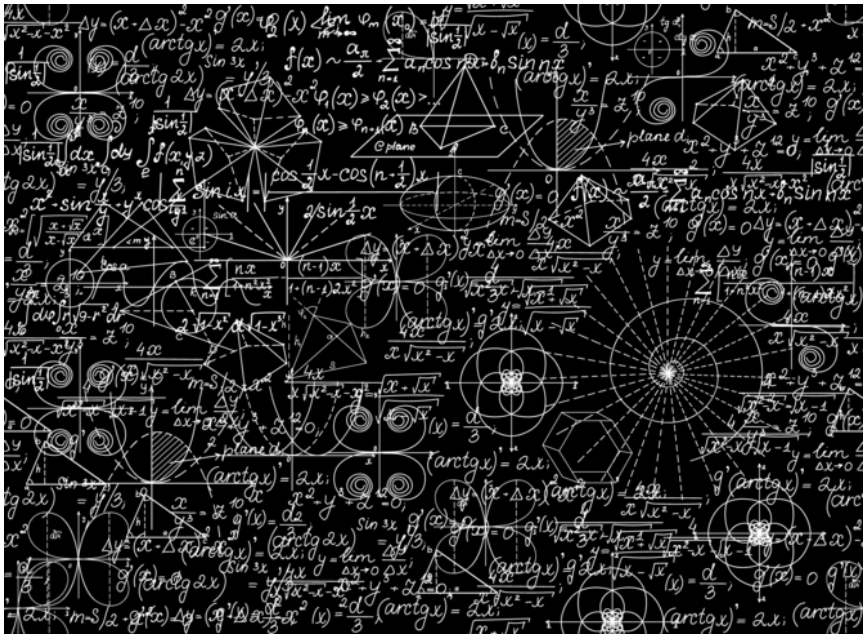
Yes! (if one-way functions exist)

Goldwasser-Micali-Rackof ['85]

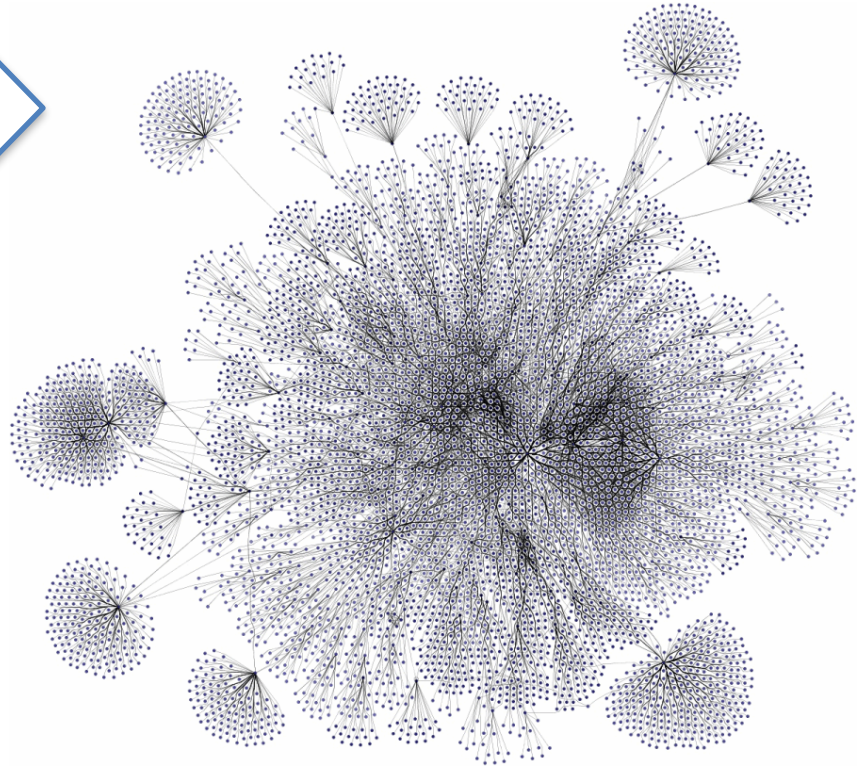
Goldreich-Micali-Wigderson ['91]

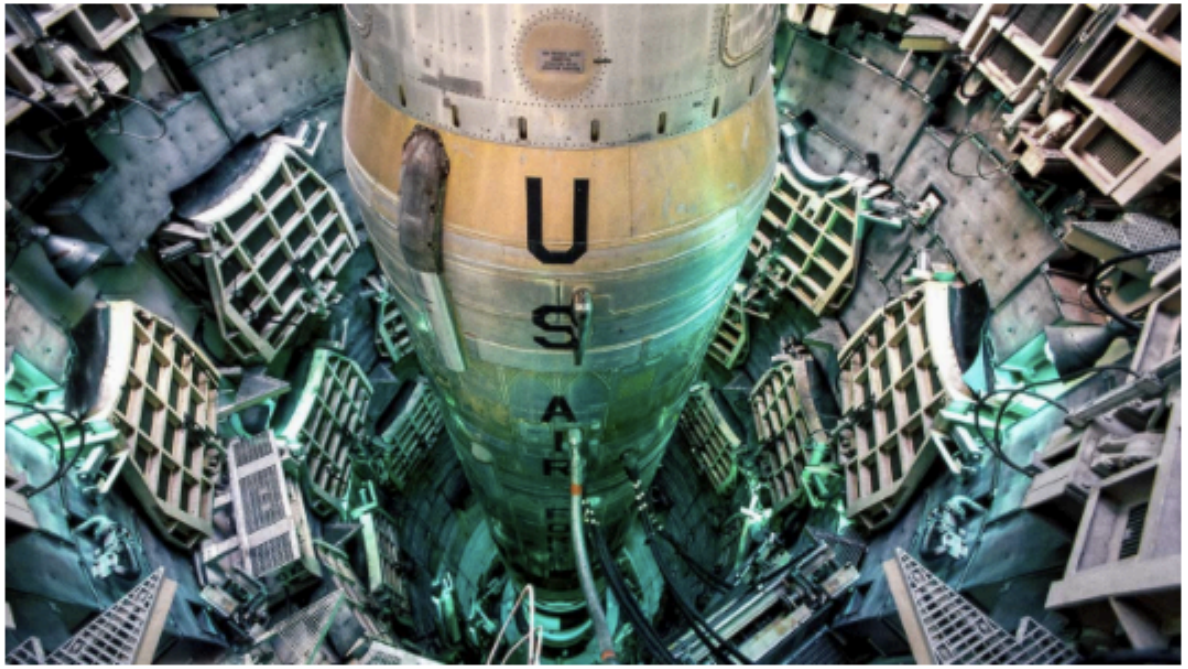
It Turns Out That…

you can turn
any proof



into a checkable
coloring of a graph





Massachusetts Institute of Technology in Cambridge physicists have proposed a new "zero knowledge" system to check the status of nuclear warheads, meaning inspectors wouldn't have to rely on less direct methods such as counting missiles.

a 'zero-knowledge' warhead inspection system

What is Computational Thinking?

- Are there any problems we can't ever solve?
- Can two strangers share secrets in the open?
- If I proved the Riemann hypothesis, could I convince my colleagues without revealing anything about the proof?
- **If white can always win in chess, can an omnipotent being quickly prove it?**

Convincing Mortals

Let's say God wants to prove His omnipotence by quickly convincing us mortals that white can always win at chess. Can he convince us?

(phrasing from Rudich and Aaronson)

Problem: way more possible games than atoms in the universe. (10^{120} vs 10^{80})

Convincing Mortals

Let's say God wants to prove His omnipotence by quickly convincing us mortals that white can always win at chess. Can he convince us?

(phrasing from Rudich and Aaronson)

by beating Kasparov?
not good enough

by beating best computers?

still not good enough

“IP=PSPACE”

By letting us ask Him some random questions!

Lund, Fortnow, Karloff, Nisan, and Shamir [‘90]

Shamir [’91]

This question isn’t as much about chess as it is about zeroes of polynomials over finite fields and error correcting codes.

Just the Tip of the Iceberg

At the boundary of math and CS lies an intellectually exciting field that has had tremendous impact on the world.

It's full of beautiful, surprising, and philosophically interesting results.

Computer Science at UIC

Mathematics and Computer Science (LAS)

Computer Science (Engineering)

Computer Engineering (Engineering)

Information & Decision Science (Business)

Mathematics and Computer Science

Mathematics and Computer Science is not a double major in two disparate fields.

MCS is a comprehensive program of study that gives you the skills and training to understand our increasingly-algorithmic world.

for more info: http://homepages.math.uic.edu/~mcs/

Why MCS?

In MCS, we teach you the programming skills you need to become a software engineer or to go into industry.

But we also focus on developing logical reasoning and on understanding the underpinnings of CS.

Our graduates are well-prepared for exciting industry positions, careers in science, and much else!

Why MSC at UIC?

We are one of the great urban research universities, and MSCS is a very strong, internationally-recognized department.

You have access to the many opportunities and resources of Chicago.

And you get to learn from (and work with) faculty at the forefront of cutting-edge research.

Thank you!

Questions?

email contacts:

Lev Reyzin (me): lreyzin@uic.edu

Florencio Diaz (Director, Advising & Outreach): fdiaz4@uic.edu

Roman Shvydkoy (Director, Undergraduate Services): shvydkoy@uic.edu