UNIVERSITY OF
EASTERN FINLAND

# A Quantum Journey on Shor's Algorithm

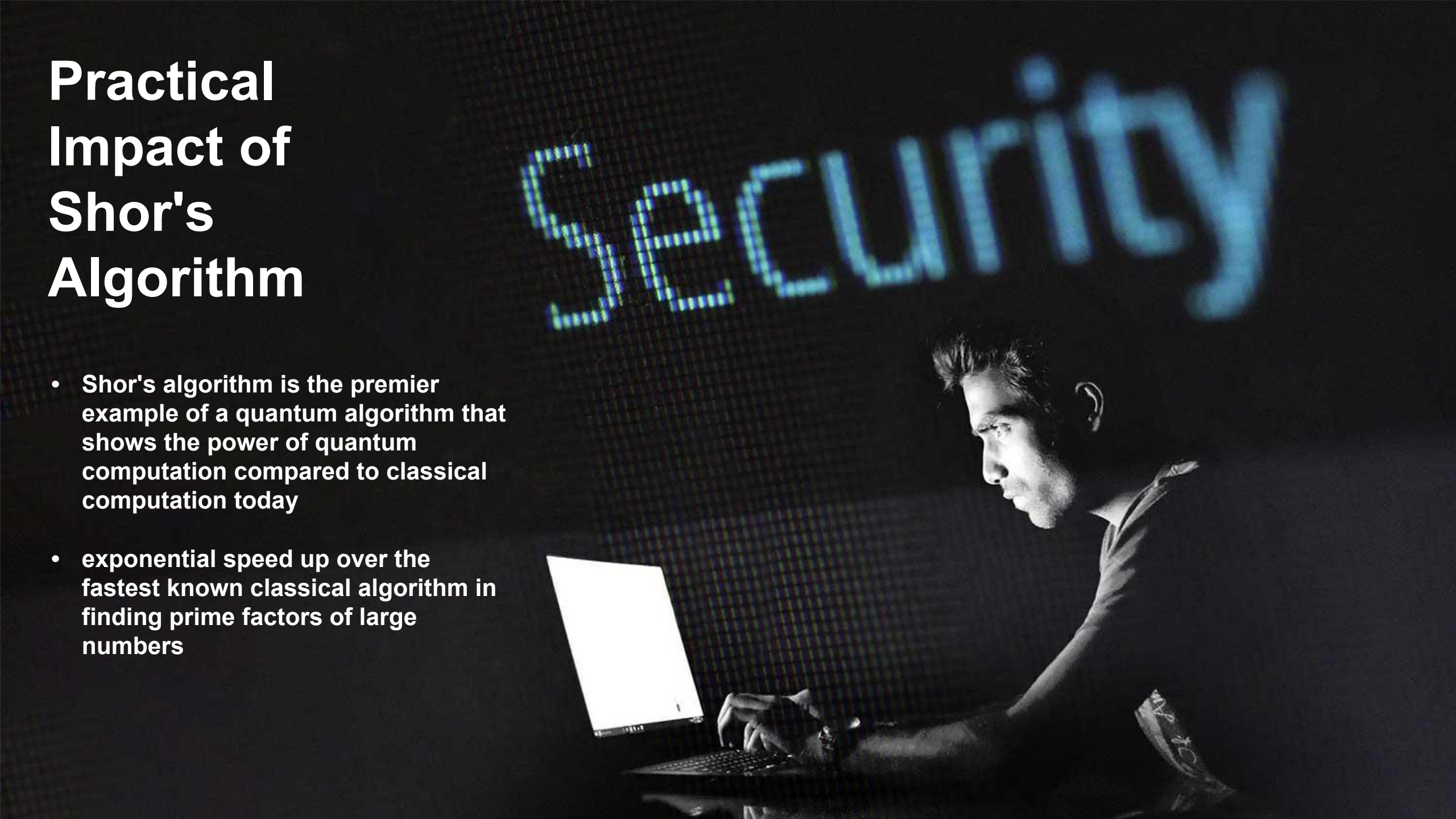
# Our Quantum Bits

| 00 | 01 | 10 | 11 |
| --- | --- | --- | --- |
| **Maiju** | **Haritha** | **Olumide** | **Kulsum** |

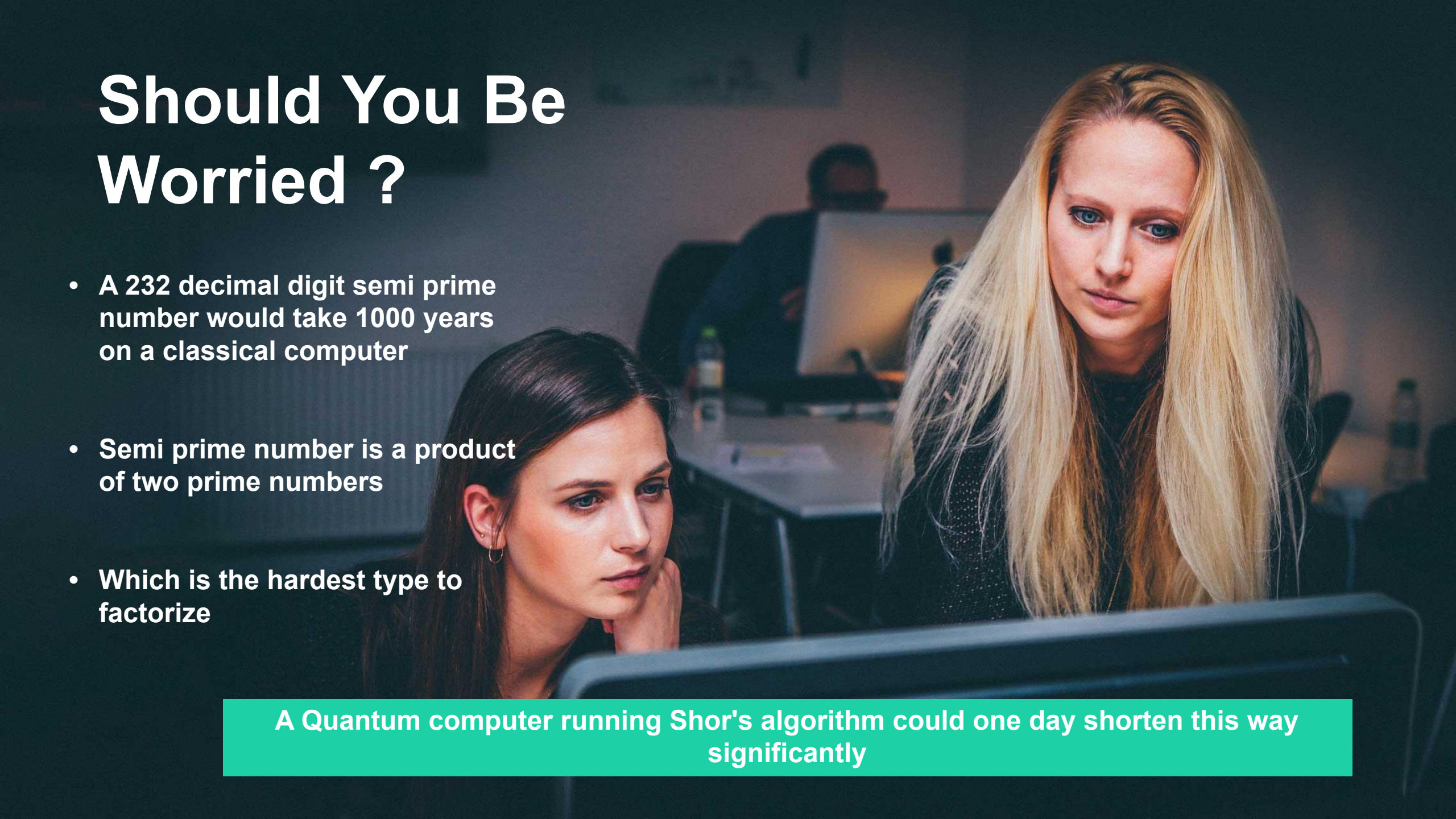
We hope you will get entangled with this presentation

# Practical Impact of Shor's Algorithm

- Shor's algorithm is the premier example of a quantum algorithm that shows the power of quantum computation compared to classical computation today
- exponential speed up over the fastest known classical algorithm in finding prime factors of large numbers



# Should You Be Worried ?

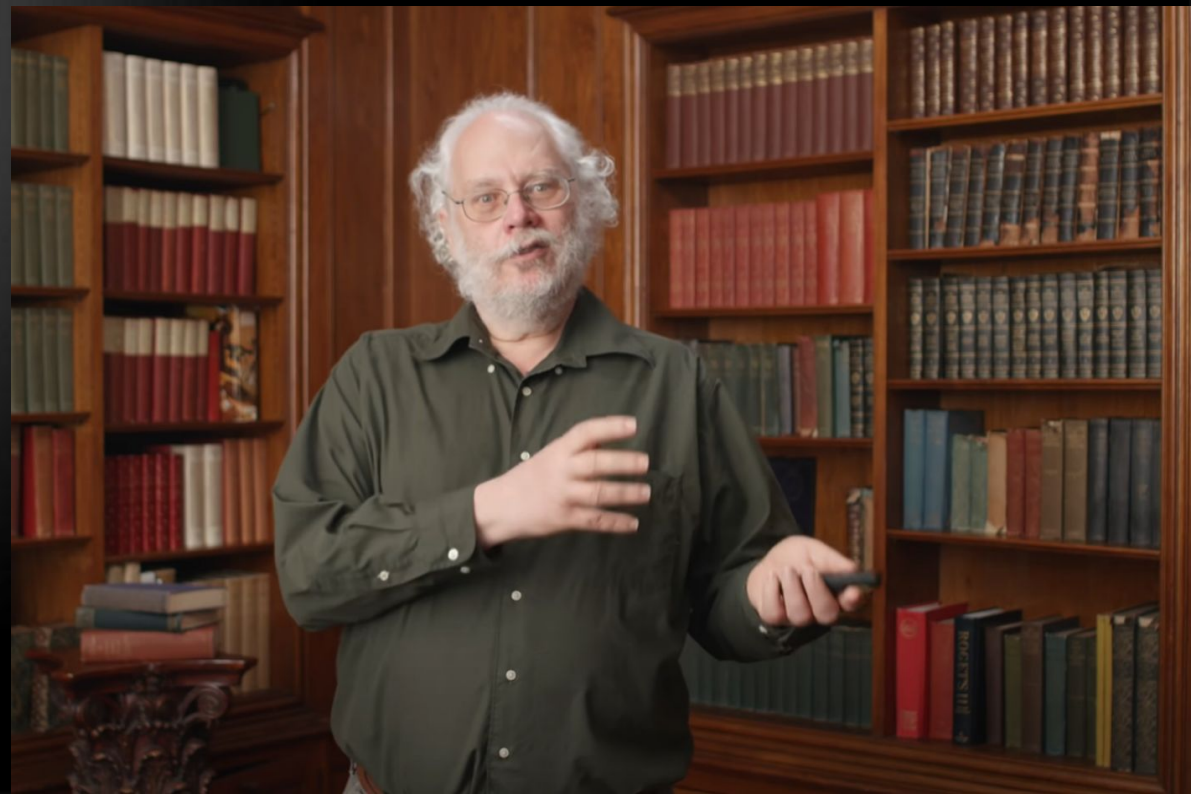


- A 232 decimal digit semi prime number would take 1000 years on a classical computer
- Semi prime number is a product of two prime numbers
- Which is the hardest type to factorize

A Quantum computer running Shor's algorithm could one day shorten this way significantly


# Road Map of Shor's Algorithm

"How it all started"



Richard Feynman

1981

**Senior At Caltech**

**Feynman's interesting lecture which Shor calls as Negative Probability using bell's theorem (which said quantum computing cannot be local and realistic)**

**Talk at Bell Labs**

**The first time Shor heard about quantum computing was in a talk that Charlie gave at Bell Labs , after BB84 in 1984**

1984

Charlie Bennett


1992

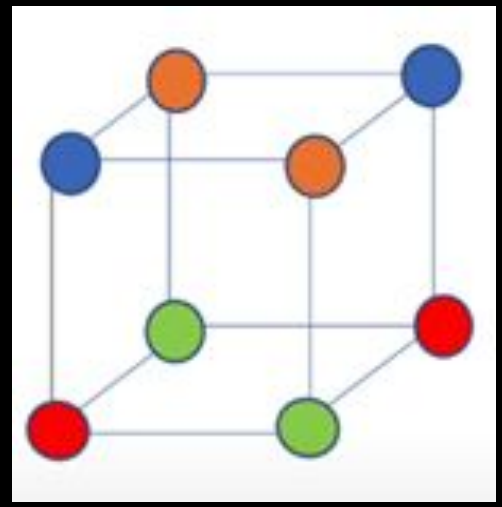
**Talk at Bell Labs**

gave his talk about Bernstein and Vazirani where he put quantum Turing machines into a mathematically rigorous framework

**Dan Simon's Paper**

he discussed about finding periods on the vertices of a high dimensional cube. So Simon's problem was given a function like this which we only can access as an oracle find the period. And the way he did was by applying what is essentially a Fourier transform over a binary vector space

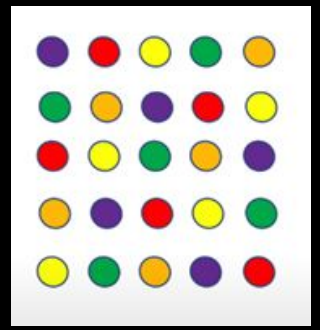
1994

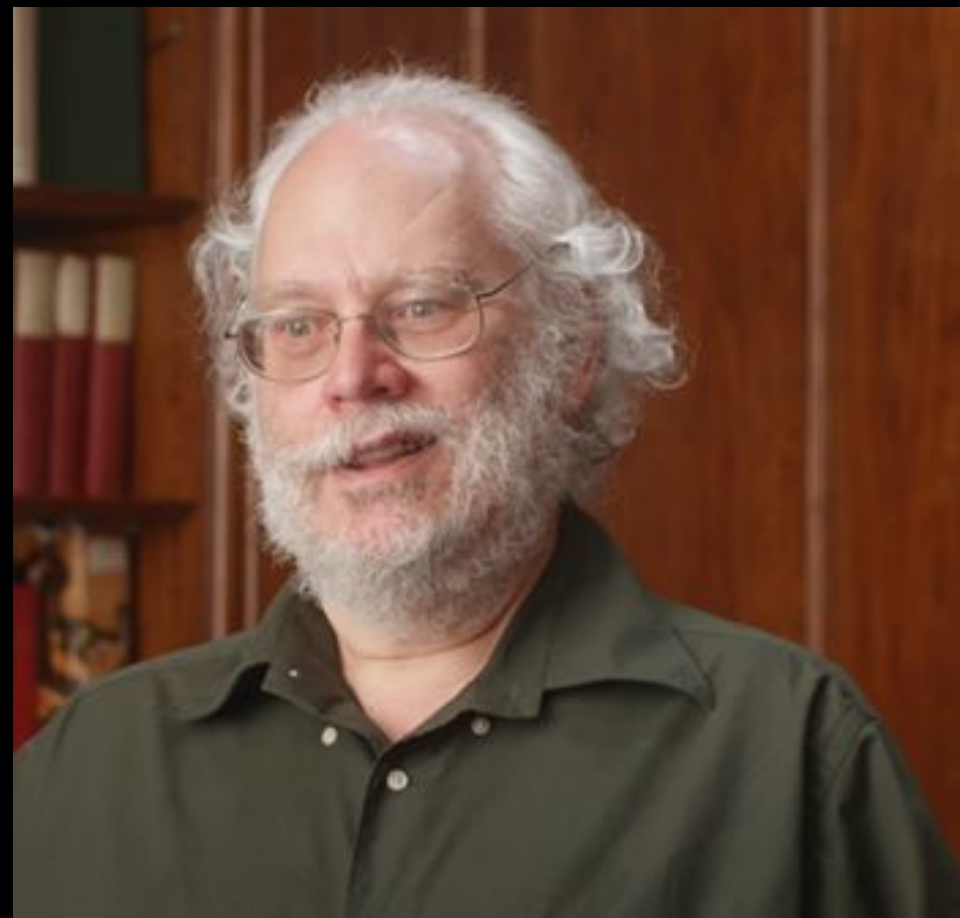


1994

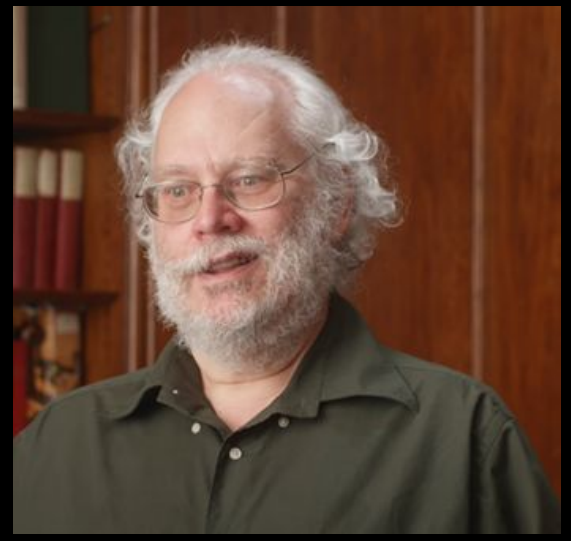
## Shor's Algorithm

Shor looked into whether he can solve this discrete log problem. Now the function is not a high dimensional cube but on a very large torus. There is a period here if you move two vertices right and one down you will find the same color and if you can find the period on a large one , where you can query the colors of the vertices by an oracle you can solve discrete log






After that Umesh Vazirani Called Shor and said "I hear you can factor on a quantum computer , tell me how it works"



# How the Shor's algorithm works?

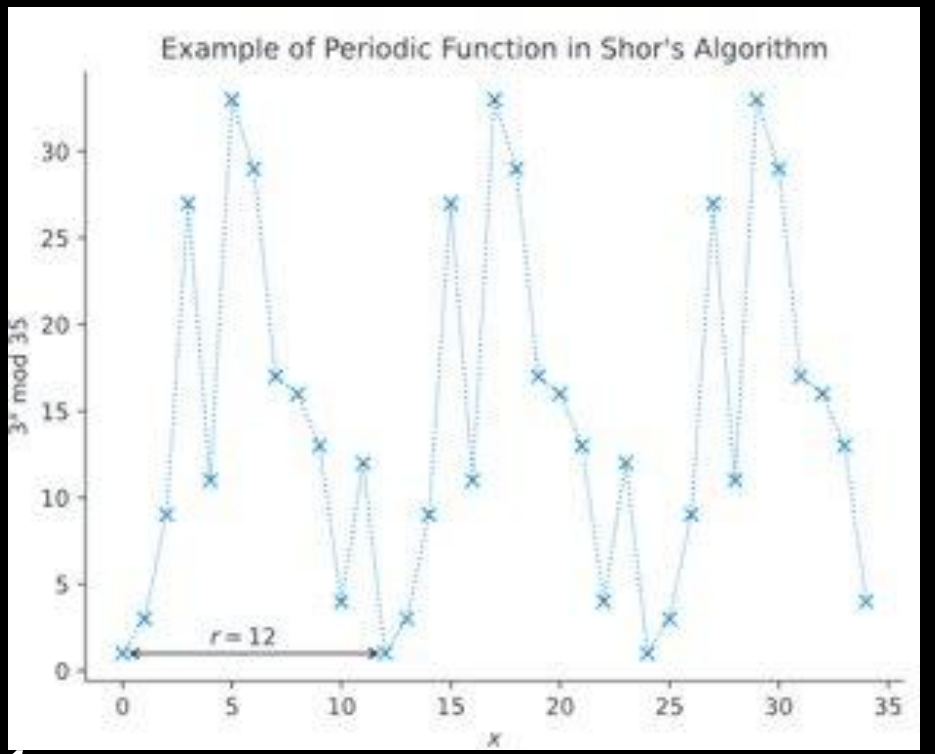
- We want to find the factors of a number *N*
- The idea is to first just start somewhere
  - Make a guess *g*
- It's easy to find the greatest common divisor of *N* and *g*
  - And if we find one then we're done!
- Unfortunately, finding a number *g* that has a common divisor with *N* other than 1 is very difficult...

- Numbers with a greatest common divisor of 1 are called coprime
- If g and N are coprime, we know the following

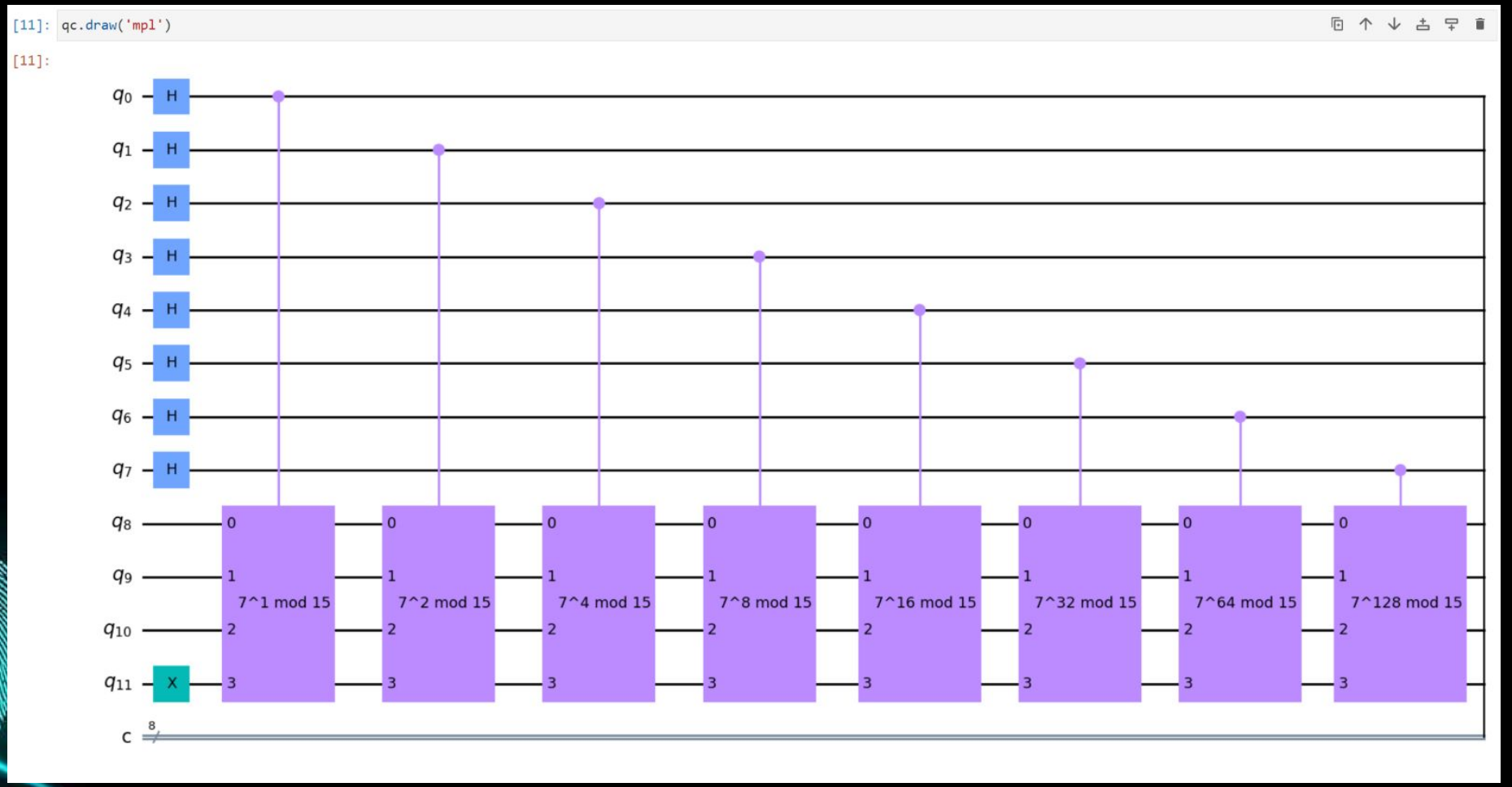
  $g^p = m \cdot N + 1$

  $(g^{p/2} + 1)(g^{p/2} - 1) = m \cdot N$
- We can use *g* to find factors of N!
- We just need to find a good value *p*

- If *g* and *N* are coprime, we know the following
  $g^p = m \cdot N + 1$
- We want to find p so that
  $g^p \bmod N = 1$
- The modulus operation repeats the same result after a certain period
- There is a *p* so that
- $g^p \bmod N = g^{2p} \bmod N = g^{3p} \bmod N \ldots = 1$
- There are many possible choices of *p*, this is where the quantum part starts




# Use Hadamard gates to create superpositions of possible values of *p*



1. Use hadamard operation to create superpositions of possible values of p
   - Get the results of the modulus operation as the output
2. To find the period after which the operation returns 1 we use the quantum fourier transform
   - This is now our value *p* we wanted to find

## Example

$N = 15$, $g = 7$, $p = 4$

$(g^{p/2} + 1)(g^{p/2} - 1) = m \cdot N$

$50 \cdot 48 = m \cdot 15$

Greatest common divisor of 50 and 15 = 5

Greatest common divisor of 48 and 15 = 3

# Thank you

*This presentation has been measured!!!*