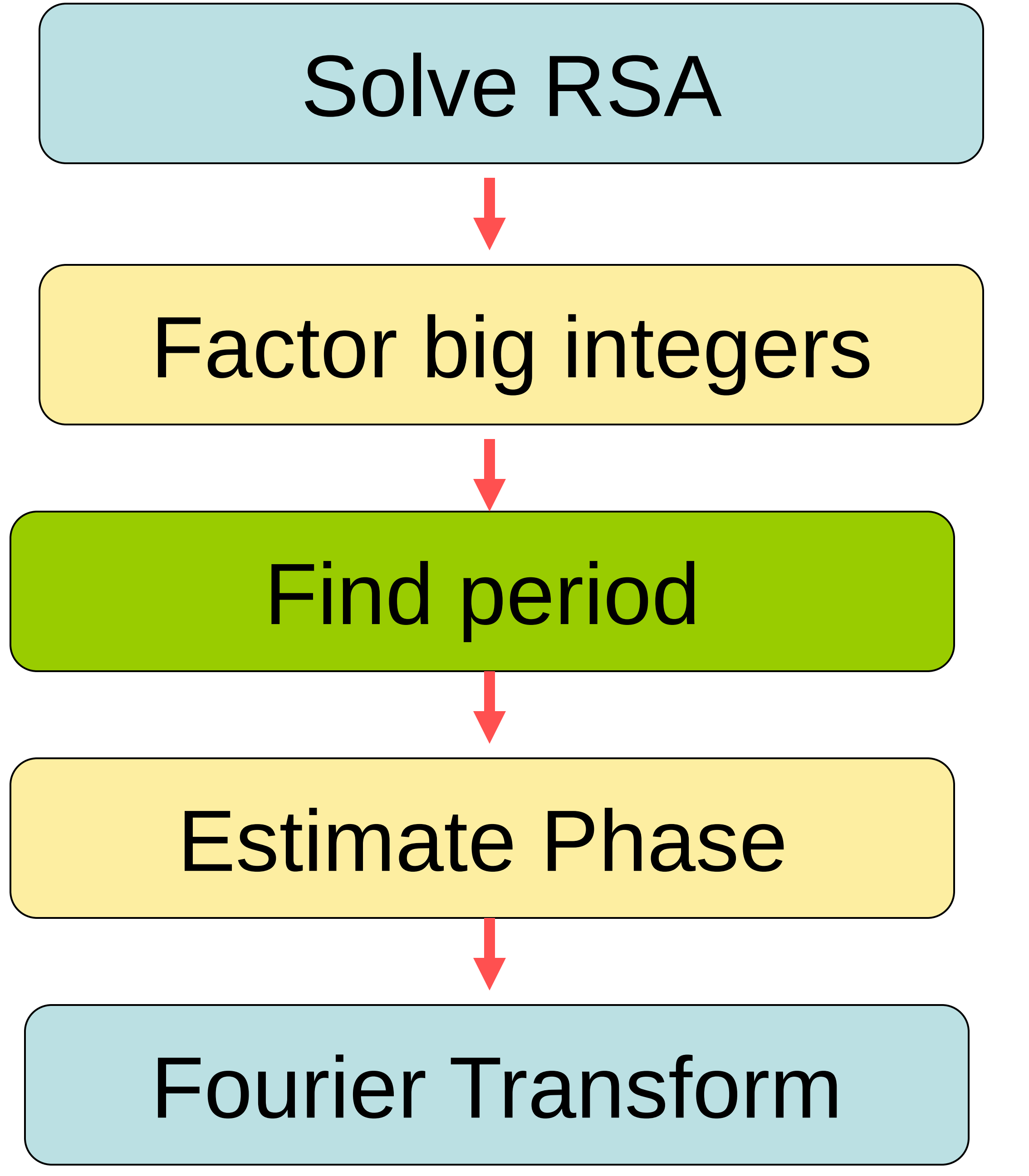
Shor Algorithm (continued)

Use of number theory and reductions

Anuj Dawar

Reductions


Solve RSA
Factor big integers
Find period
Estimate Phase
Fourier Transform

RCA ENCRYPTION

RSA encryption

- Named after Rivest, Shamir and Adleman, who came up with the scheme

$$m_1 \times m_2 = N$$

Primes

- Based on the ease with which N can be calculated from m_1 and m_2
- And the difficulty of calculating m_1 and m_2 from N

Easy to multiply but difficult to factor big integers.

RSA encryption

- N is made publicly available, and is used to encrypt data
- m_1 and m_2 are the secret keys which enable you to decrypt the data
- To crack the code, a code-breaker needs to factor N
- Best current cracking method on a classical computer
  - Number field sieve
  - Requires $\exp(O(n^{1/3} \log^{2/3} n))$
  - n is the length of N

Review of Number Theory

Shor knows number theory and uses it!!!

1. In many cases, we can use the knowledge from other areas of research in a new and creative way.

2. You do not have to invent everything from scratch. You just reuse something that was invented by other people.

3. If the two areas are not obviously linked, your invention can be very important.

4. This is exactly what was done by Shor.

1. We introduced modular arithmetic in last lecture as a general tool for algorithms and hardware

2. Now we will show how creatively Shor used it in his algorithm.

A little number theory

Assume:

$$m_1 \times m_2 = N$$

Modular Arithmetic

$$a \equiv b \bmod N$$

Simply means

$$a = b + kN$$

k is any integer

and $b < N$

Smallest

$$a^r \equiv 1 \bmod N$$

Co-prime

$$\gcd(a, N) = 1$$

Greatest Common Divisor

No factors in common!

We want to find the smallest r such that the above is true

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

Consider the equation

We want to find the smallest r such that the above is true

$$y^2 \equiv 1 \bmod N$$

$$y^2 - 1 \equiv 0 \bmod N$$

$$(y+1)(y-1) \equiv 0 \bmod N$$

$$(y+1)(y-1) = kN$$

Now we substitute m_1 * m_2 for N

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

Greatest common denominator

$$(y+1)(y-1) = k m_1 m_2$$

$$\gcd(y+1, N) = N$$

$$\gcd(y-1, N) = 1$$

Trivial solutions

More interesting case

$$\gcd(y+1, N) = m_1$$

$$\gcd(y-1, N) = m_2$$

- gcd can be calculated very efficiently
- Euclid's algorithm
- 300 BC

A little number theory

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

- If we can find r
- And the r is even
- Then

$$y^2 \equiv 1 \bmod N$$

$$m_1 = \gcd(a^{r/2} + 1, N)$$
$$m_2 = \gcd(a^{r/2} - 1, N)$$

- Provided we don't get trivial solutions

We want to find the smallest r such that the above is true → Finding the smallest period r

A little number theory

But we had some additional assumptions on last slide, what if not satisfied?

$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

- What about the ifs and buts ?!?

Theorem:

Let $N = m_1 m_2$, where m_1 and m_2 are prime numbers not equal to 2. Suppose a is chosen at random from the set $\{a : 1 < a < N, \gcd(a,N) = 1\}$. Let r be the order of y mod N. Then the probability

$$\text{Prob}(r \text{ is even and non-trivial}) \geq \frac{1}{2}$$

Proof: long, boring and complicated

Do not worry now, we are not mathematicians

A little number theory

So now we are quite optimistic!

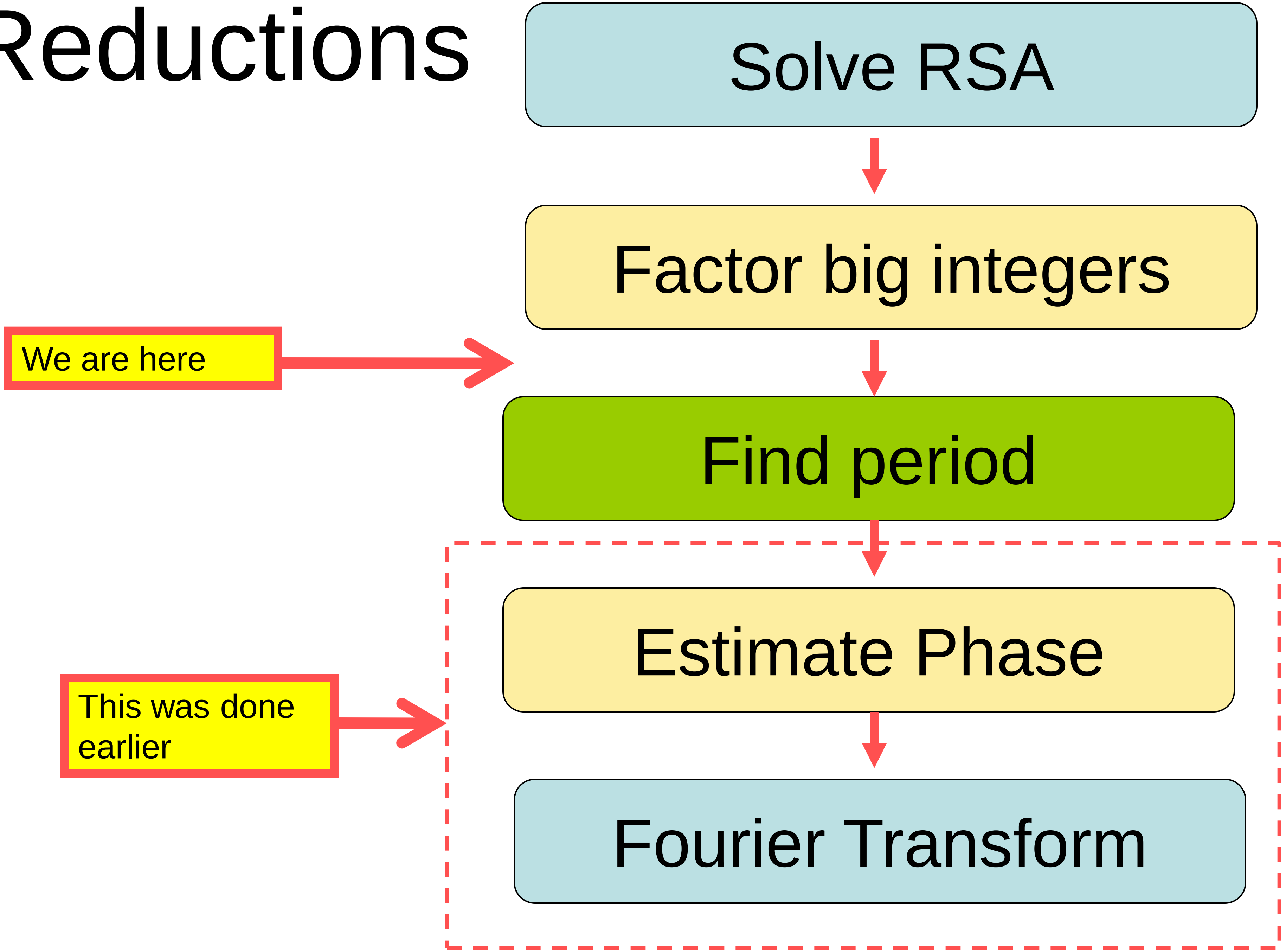
$$m_1 \times m_2 = N \quad \Leftrightarrow \quad a^r \equiv 1 \bmod N$$

- Finding r is equivalent to factoring N
- Why can't we use a classical computer to find r?
  - It takes $O(2^n)$ operations

So now what remains is to be able to find period, but this is something well done with spectral transforms.

Exercise: Using the reduction of factoring to order-finding, and the fact that 10 is co-prime to 21, factor 21

Reductions


Solve RSA
Factor big integers
We are here
Find period
This was done earlier
Estimate Phase
Fourier Transform

Going Back to Phase Estimation

We will use phase estimation to find period

Choosing the operator U

1. It requires modulo multiplication in modular arithmetic
2. Not trivial
3. Potential research how to do this efficiently

Choosing a U

$$a^r \equiv 1 \bmod N$$

- Consider the operator,

$$U|x\rangle \to |ax \bmod N\rangle$$

- As a and N are co-prime, this operator is unitary
- Can be efficiently implemented on a quantum computer
- What about U^2, U^4, U^8, ..., U^{2^j}

$$U^2|x\rangle \to |a^2x \bmod N\rangle$$

Choosing the initial state for operator U

1. In general not easy
2. But hopefully we find a special case
3. Potential research how to do this efficiently for arbitrary cases

Choosing an initial state

$$a^r \equiv 1 \bmod N$$

- Consider the state,

$$|\psi_1\rangle = \sum_{j=0}^{r-1} e^{\frac{-2\pi ij}{r}} |a^j \bmod N\rangle$$

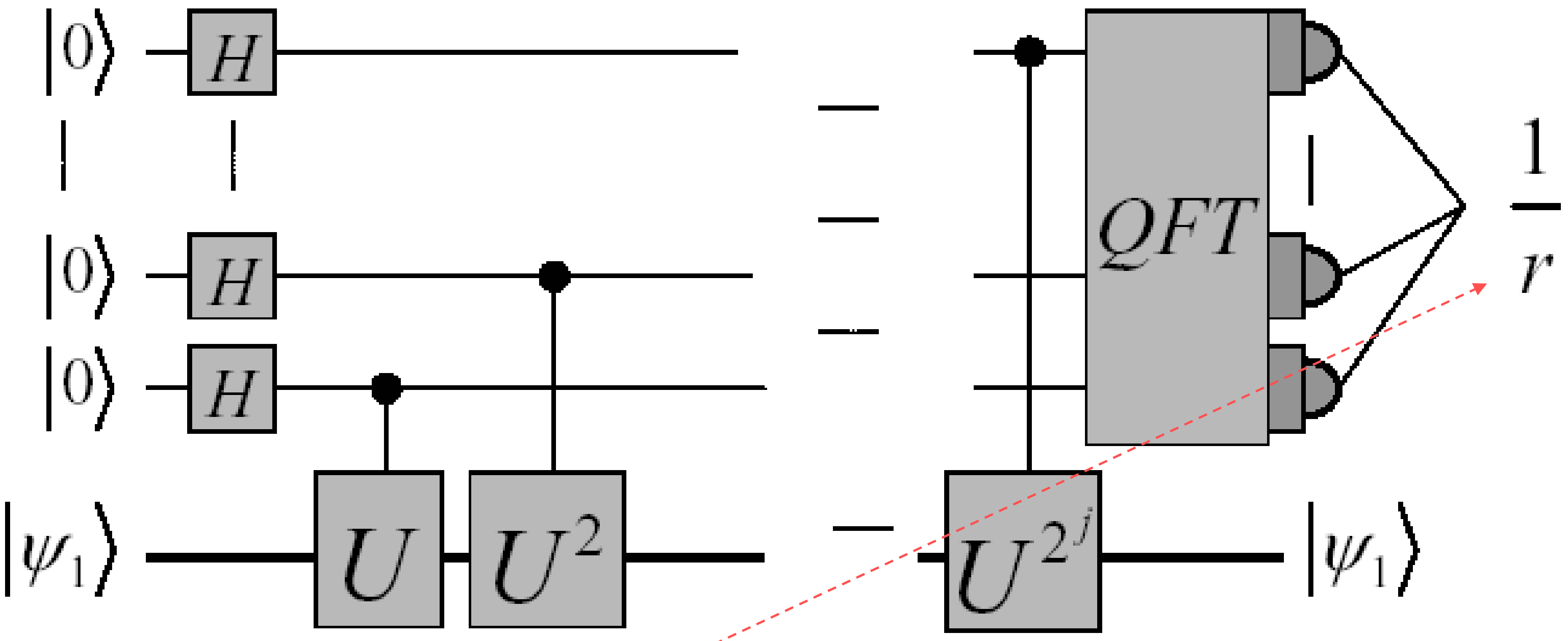
- $|\psi_1\rangle$ is an eigenstate of U, with eigenvalue

$$e^{2\pi i(\frac{1}{r})}$$

Phase is 1/r

- Therefore, if we could prepare $|\psi_1\rangle$, we can use the PE algorithm to efficiently find r, and hence factor N.

Choosing an initial state



- Therefore, if we could prepare $|\psi_1\rangle$, we can use the PE algorithm to efficiently find r, and hence factor N.

Now the problem is reduced to creation of certain quantum state.
We published papers – see David Rosenbaum

Choosing an initial state

$$a^r \equiv 1 \bmod N$$

- Consider the states,

$$\left|\psi_1\right\rangle = \sum_{j=0}^{r-1} e^{\frac{-2\pi ij}{r}} \left|a^j \bmod N\right\rangle$$

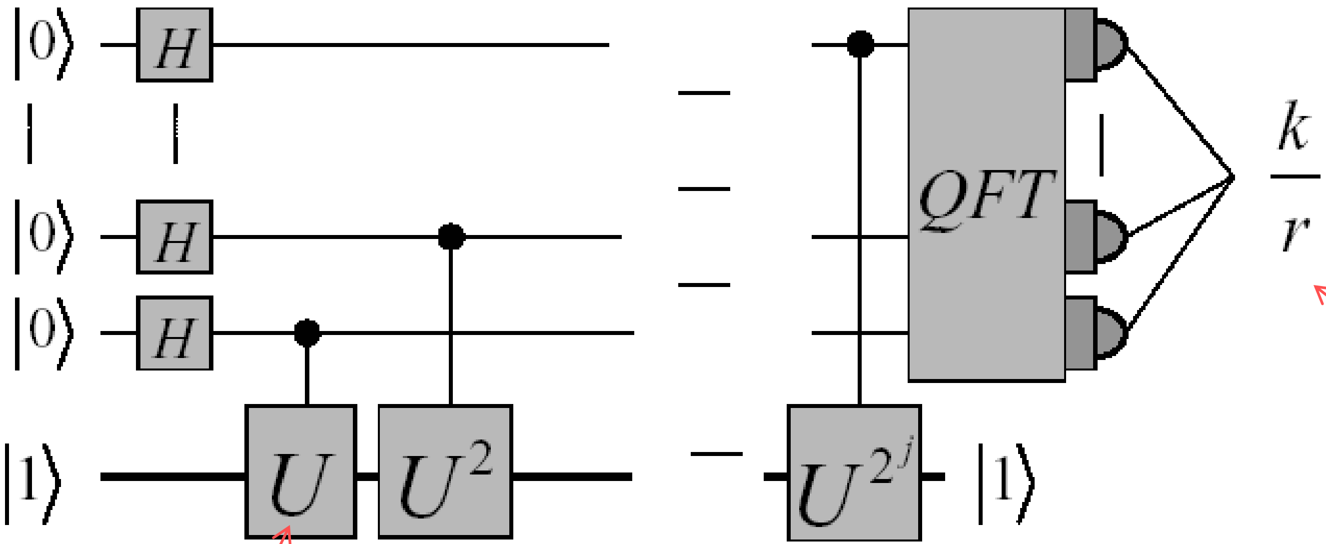
$$k \in \{1, \ldots r\}$$

- $\left|\psi_k\right\rangle$ is an eigenstate of U, with eigenvalue

$$e^{2\pi i(\frac{k}{r})}$$

Exercise: Show $\left|1\right\rangle = \sum_{k=1}^{r} \left|\psi_k\right\rangle$

Final circuit for period finding



We find this

Easy initialization

$$|1\rangle = \sum_{k=1}^{r} |\psi_k\rangle$$

Number to be factorized

U |x⟩ → |a*x mod N ⟩

$a^r = 1 \bmod N$

Now we use a classical computer.

1. Therefore, using the QPE algorithm, we can efficiently calculate

$$\frac{k}{r}$$

where k and r are unknown

2. If k and r are co-prime, then canceling to an irreducible fraction will yield r.

3. If k and r are not co-prime, we try again.

Summary of Shor Algorithm

1. We want to find **$m_1 * m_2 = N$** where N is the number to factorize

2. We prove that this problem is equivalent to solving

$$a^r = 1 \bmod N$$

3. We use the QPE circuit initialized to $|0\rangle\ |1\rangle$
4. We calculate each of the circuits **$U, U^2, \ldots U^{2\wedge 2n}$**
5. We apply the Quantum Phase Estimation Algorithm.
6. We use standard computer for verification and we repeat QPE if required.