CS 258: Quantum Cryptography

Mark Zhandry

Previously…

Composite systems

Suppose we had two states

$$|\psi\rangle = \sum_i \alpha_i |i\rangle \qquad\qquad |\phi\rangle = \sum_j \beta_j |j\rangle$$

The two together also form a quantum state on a larger system

$$|\psi\rangle \otimes |\phi\rangle = \sum_{i,j} \alpha_i \beta_j |i, j\rangle$$

Also write as $|\psi\rangle|\phi\rangle$

$\{|i, j\rangle\}_{i,j}$ is computational basis for composite system

Composite systems

Often convenient to name systems

$|\psi\rangle_{\mathcal{A}}$

System $\mathcal{A}$ is in state $|\psi\rangle$

$|\phi\rangle_{\mathcal{B}}$

System $\mathcal{B}$ is in state $|\phi\rangle$

$|\zeta\rangle_{\mathcal{AB}}$

Joint system $\mathcal{AB}$ is in state $|\zeta\rangle$

Entanglement

$$|\zeta\rangle_{\mathcal{AB}} = \sum_{i,j} \gamma_{i,j} |i\rangle_{\mathcal{A}} |j\rangle_{\mathcal{B}}$$

Separable states: $|\zeta\rangle_{\mathcal{AB}} = |\psi\rangle_{\mathcal{A}} |\phi\rangle_{\mathcal{B}}$

Most states are **not** separable. In such case, we say $\mathcal{A}, \mathcal{B}$ are **entangled**

Cloner: unitary U such that

$$U|\psi\rangle|0\rangle|0\rangle = |\psi\rangle|\psi\rangle|\tau_\psi\rangle \quad \text{for any} \quad |\psi\rangle$$

$|\tau_\psi\rangle$ is arbitrary state

No-cloning Thm: For dimension >1, there is no cloner

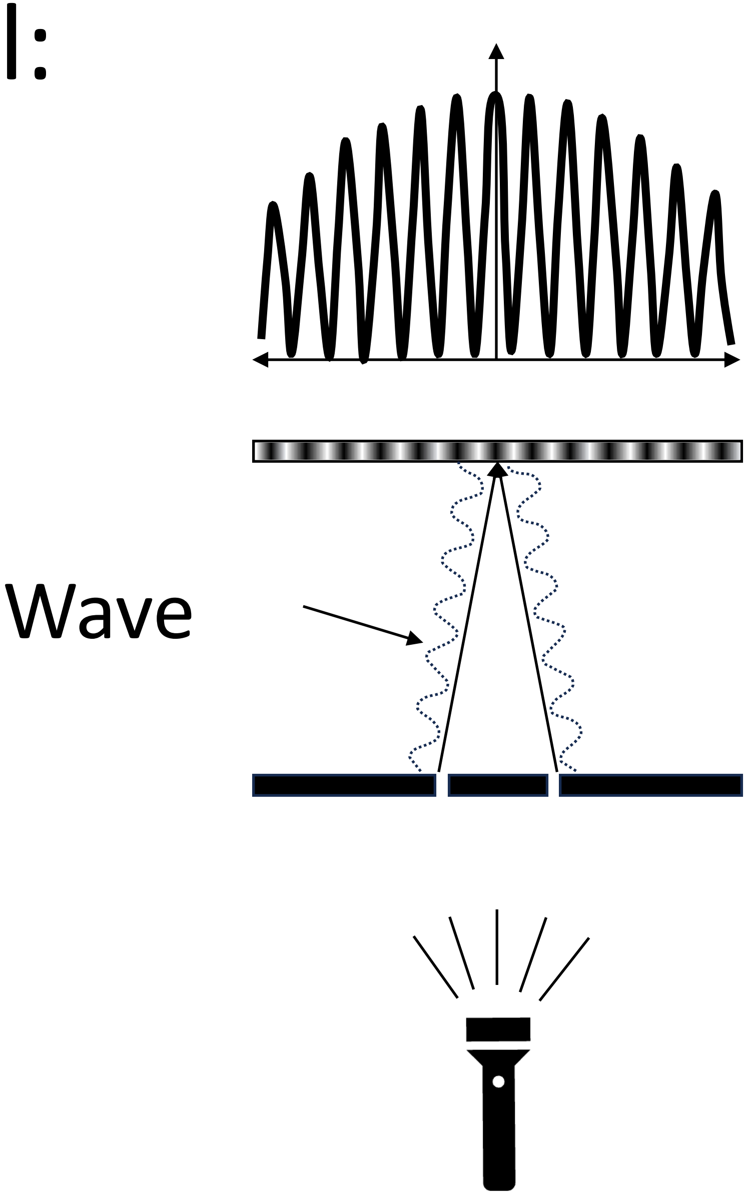
Today: quantum computing!

What quantum computing is **not**

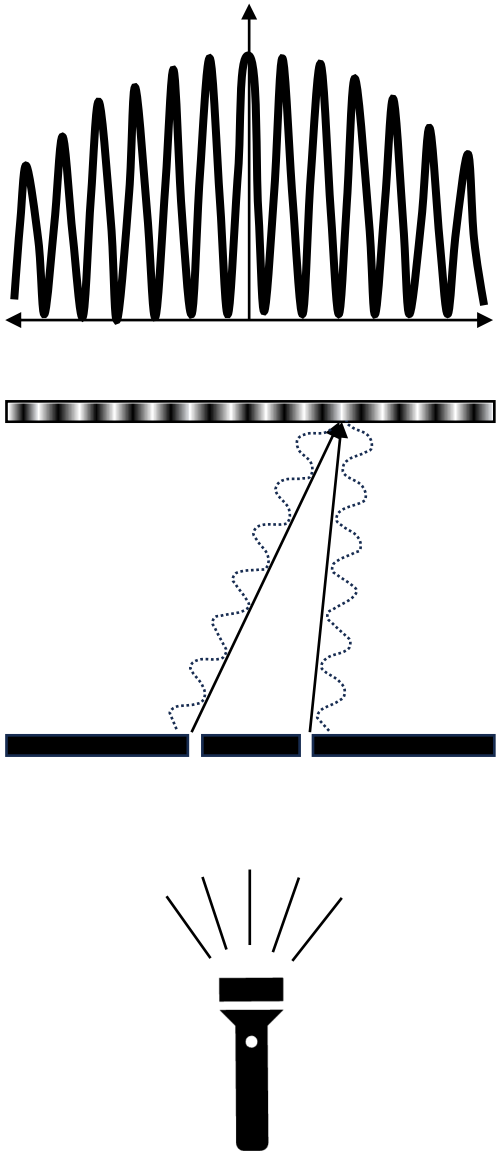
Wrong idea: a quantum computer can be in a superposition over all exponentially-many states of a classical computer. Therefore, it can try all possibilities at the same time

Reason its wrong: Same can be said of our model of classical statistical mechanics, but clearly that doesn’t offer any speedup. Any actual speedups need to use something inherently quantum

Recall:



Constructive Interference



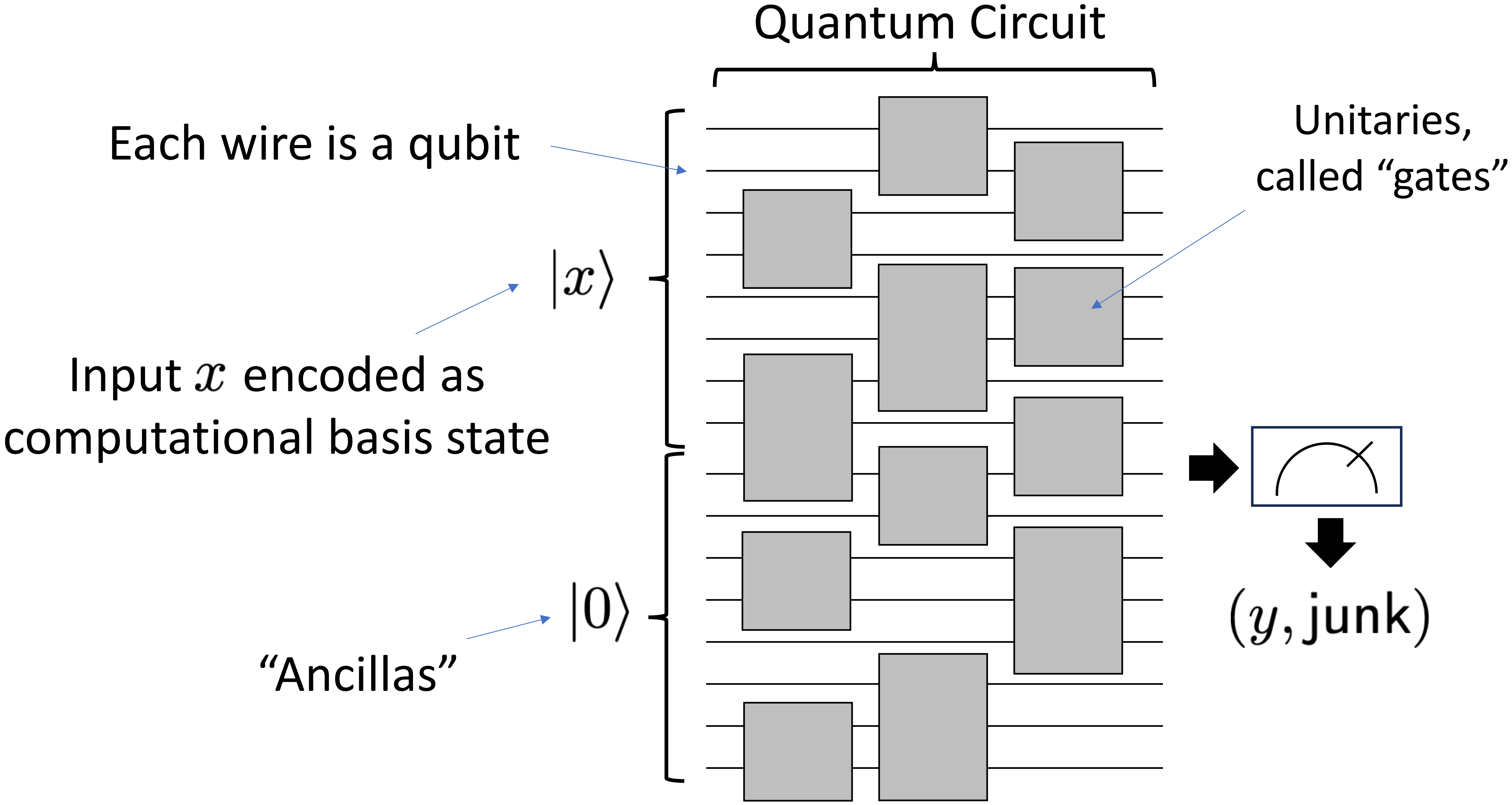
Destructive Interference

In a nutshell, quantum computing is about using **interference** so that different computational paths leading to the correct answer constructively interfere, and computational paths leading to the wrong answers destructively interfere. The result is that the right answer is achieved with higher probability than what is possible classically

Obtaining the right answer with higher probability often means being able to obtain the right answer *faster*

Qubit

A qubit is just a 2-dimensional quantum system over $|0\rangle$ and $|1\rangle$



Quantum Circuit

Each wire is a qubit

Unitaries, called “gates”

$|x\rangle$

Input x encoded as computational basis state

$|0\rangle$

“Ancillas”

(y, junk)

Each quantum circuit defines a (potentially probabilistic) process to convert an input x into an output y

Our goal: design an “efficient” quantum circuit to solve some task much faster than what is possible classically

BQP = “Bounded-error Quantum Polynomial time”

Efficient quantum circuits

Many ways to define that (fortunately) turn out to be equivalent. For this class, the following definition suffices

- Circuit comprised of polynomially-many (in $|x|$) gates
  - Each gate is either a 1-qubit unitary, or

$$\mathsf{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad \mathsf{CNOT}|a, b\rangle = |a, b \oplus a\rangle$$

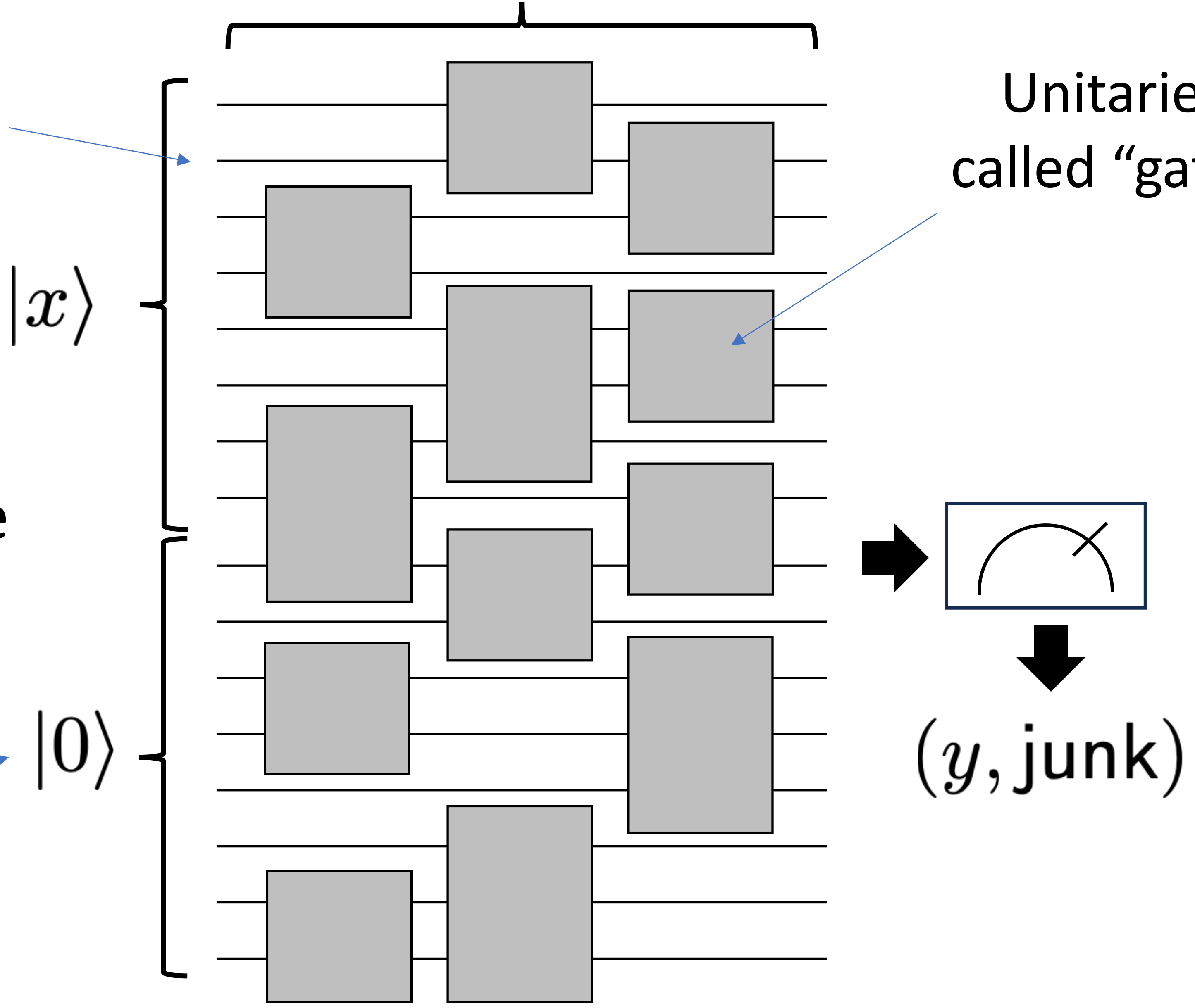
- Classical description of circuit computable by classical machine in polynomial-time from only the input length

For convenience, we will typically allow quantum algorithms to make intermediate measurements as well. It turns out that this model and the model where are measurements are at the end are essentially equivalent.

Challenges with quantum computing

- Gates are unitary → no fanout, all gates reversible
- No-cloning theorem → cannot copy internal state of algorithm (no assigning quantum variables and then using them twice)
- Observer effect → even looking at internal state changes it (no reading quantum variables without changing them)
- Entanglement → changing a variable changes other variables

Quantum Circuit



Each wire is a qubit

Unitaries,
called “gates”

Input x encoded as
computational basis state

“Ancillas”

Quantum >= Classical

Efficient classical circuits can be mapped to efficient quantum circuits with minimal overhead

But even specifying what this means requires some care

$$f : \{0,1\}^n \rightarrow \{0,1\}^m$$

Might not be injective. How to encode as unitary?

Quantum >= Classical

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle$$

Quantum >= Classical

Actually allow for workspace/ancilla qubits that get returned to 0:

$$U_f|x, y, 0^w\rangle = |x, y \oplus f(x), 0^w\rangle$$

It turns out that if we have an efficient classical circuit for f, we can construct an efficient quantum circuit for U_f (see homework)

Phase Kickback

Let $f : \{0,1\}^n \to \{0,1\}$. Define

$$P_f|x, 0^w\rangle = (-1)^{f(x)}|x, 0^w\rangle$$

Phase Kickback

$$\begin{aligned}
U_f|x\rangle|-\rangle|0^w\rangle &= \frac{1}{\sqrt{2}}\sum_b U_f|x, b, 0^w\rangle(-1)^b \\
&= \frac{1}{\sqrt{2}}\sum_b |x, b \oplus f(x), 0^w\rangle(-1)^b \\
&= \frac{1}{\sqrt{2}}\sum_{b'} |x, b', 0^w\rangle(-1)^{b'\oplus f(x)} \\
&= |x\rangle|-\rangle|0^w\rangle(-1)^{f(x)}
\end{aligned}$$

$$b' = b \oplus f(x)$$

$$\Rightarrow P_f = [\mathbf{I} \otimes (\mathbf{XH}) \otimes \mathbf{I}]U_f[\mathbf{I} \otimes (\mathbf{HX}) \otimes \mathbf{I}]$$

Grover’s Algorithm

Unstructured search

Setup: we have a function $f : \{0,1\}^n \rightarrow \{0,1\}$ with the promise that there is some x such that $f(x) = 1$

Our goal: find such an x

Captures tons of problems, in fact all of **NP** search problems: $f(x) =$ “is x a satisfying assignment for the Boolean formula C”

In particular, a general solution for this problem would break almost all classical cryptography

The Classical Setting

If f itself is efficient, and **P = NP**, the this problem can be solved in polynomial time

However, best known algorithms, even if f is efficient, cannot do much better than just trying f on all inputs (that is, brute force)

Not hard to see that such algorithms must evaluate f for roughly 2^n times

Thm: There exists a quantum algorithm that performs $O(\sqrt{2^n})$ evaluations of U_f, and finds an x such that $f(x) = 1$ with probability $1 - O(2^{-n})$

Gover's Algorithm

(we will ignore ancillas needed to compute U_f, P_f)

1. Initialize system to $|+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$

2. Repeat the following T times:

- Apply P_f
- Apply $\mathbf{H}^{\otimes n}$
- Apply P_Z
- Apply $\mathbf{H}^{\otimes n}$

$$Z(x) = \begin{cases} 0 & \text{if } x = 0^n \\ 1 & \text{if } x \neq 0^n \end{cases}$$

3. Measure

Analysis

Suppose there is a unique x^* s.t. $f(x^*) = 1$

$$P_f|x^*\rangle = -|x^*\rangle$$

$$\begin{aligned} P_f|+\rangle^{\otimes n} &= P_f \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \\ &= \frac{1}{\sqrt{2^n}} \left(-|x^*\rangle + \sum_{x \neq x^*} |x\rangle \right) \\ &= \frac{1}{\sqrt{2^n}} \left(-2|x^*\rangle + \sum_{x} |x\rangle \right) \\ &= |+\rangle^{\otimes n} - \frac{2}{\sqrt{2^n}}|x^*\rangle \end{aligned}$$

Analysis

$$\begin{aligned}
\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n} |+\rangle^{\otimes n} &= \mathbf{H}^{\otimes n} P_Z |0^n\rangle \\
&= \mathbf{H}^{\otimes n} |0^n\rangle \\
&= |+\rangle^{\otimes n}
\end{aligned}$$

Analysis

$$\begin{aligned}
\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n} |x^*\rangle &= \mathbf{H}^{\otimes n} P_Z \frac{1}{\sqrt{2^n}} \sum_x (-1)^{x \cdot x^*} |x\rangle \\
&= \mathbf{H}^{\otimes n} \frac{1}{\sqrt{2^n}} \left(|0^n\rangle - \sum_{x \neq 0^n} (-1)^{x \cdot x^*} |x\rangle \right) \\
&= \mathbf{H}^{\otimes n} \frac{1}{\sqrt{2^n}} \left(2|0^n\rangle - \sum_x (-1)^{x \cdot x^*} |x\rangle \right) \\
&= \frac{2}{\sqrt{2^n}} |+\rangle^{\otimes n} - |x^*\rangle
\end{aligned}$$

Analysis

$$P_f|x^*\rangle = -|x^*\rangle$$

$$P_f|+\rangle^{\otimes n} = |+\rangle^{\otimes n} - \frac{2}{\sqrt{2^n}}|x^*\rangle$$

$$\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n}|x^*\rangle = -|x^*\rangle + \frac{2}{\sqrt{2^n}}|+\rangle^{\otimes n}$$

$$\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n}|+\rangle^{\otimes n} = |+\rangle^{\otimes n}$$

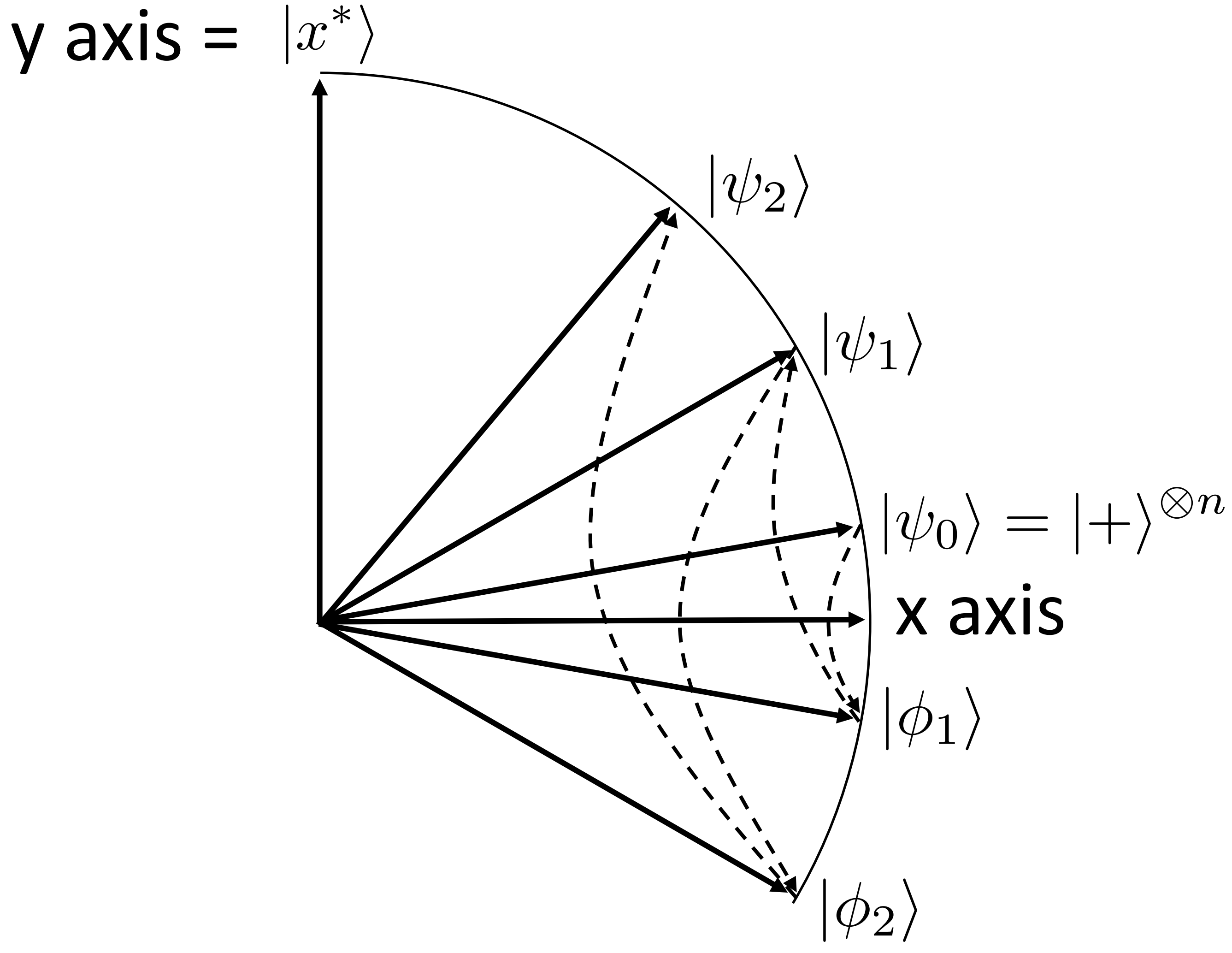
Analysis

Let S be span of $|x^*\rangle$ and $|+\rangle^{\otimes n}$

Then the state of the algorithm always stays in S

Define:

$$|\psi_0\rangle = |+\rangle^{\otimes n}$$
$$|\phi_i\rangle = P_f|\psi_{i-1}\rangle$$
$$|\psi_i\rangle = \mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n}|\phi_i\rangle$$



P_f reflects about x-axis

$\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n}$ reflects about $|+\rangle^{\otimes n}$

Analysis

Composing two reflections gives a **rotation**

Let θ_i be angle $|\psi_i\rangle$ makes with x-axis

Let τ_i be angle $|\phi_i\rangle$ makes with x-axis

P_f reflects about x-axis ➡ $\tau_i = -\theta_{i-1}$

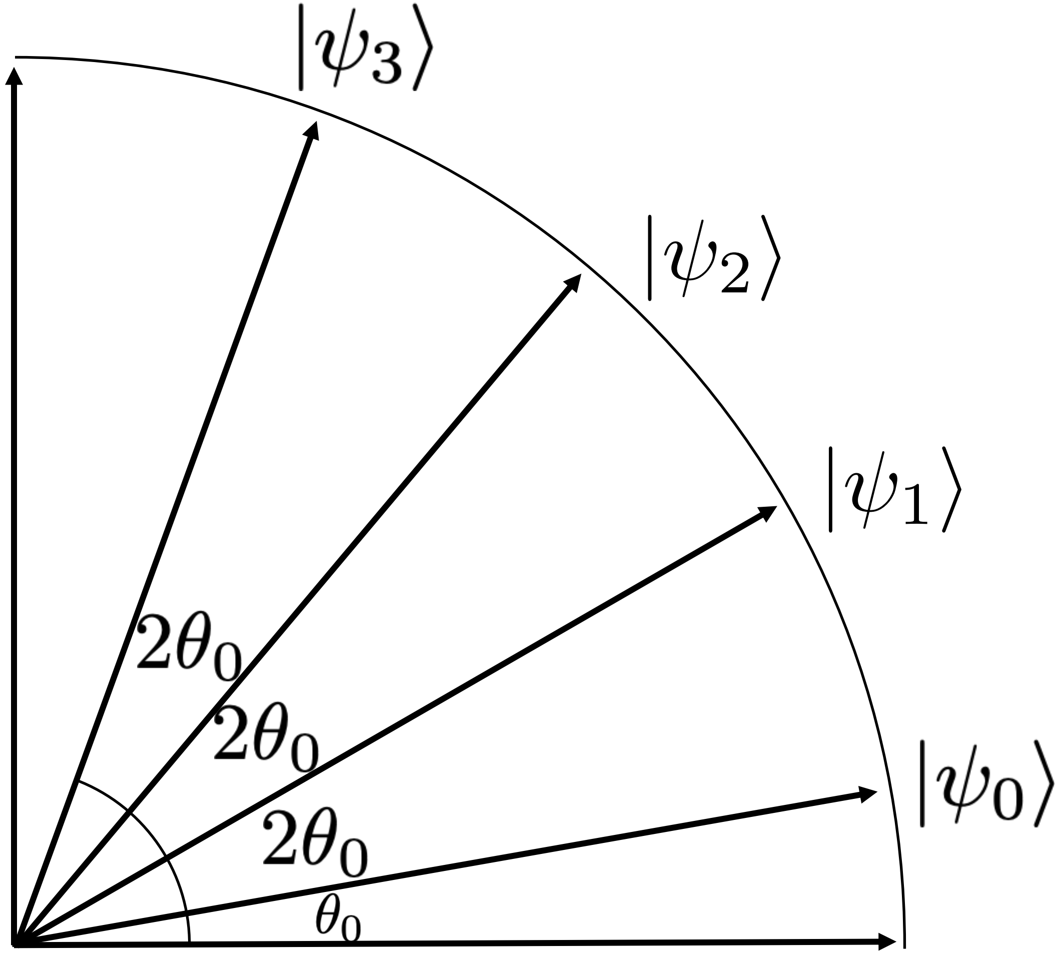
$\mathbf{H}^{\otimes n} P_Z \mathbf{H}^{\otimes n}$ reflects about $|\psi_0\rangle = |+\rangle^{\otimes n}$

➡ $\theta_i = \theta_0 - (-\theta_0 + \tau_i) = 2\theta_0 - \tau_i = 2\theta_0 + \theta_{i-1}$

Angle $|\phi_i\rangle$ makes with $|\psi_0\rangle$

Analysis

Goal: $\theta_T \approx \pi/2$



Analysis

Solve $\theta_T = (2T+1)\theta_0 \approx \pi/2$

$$\theta_0 = \sin(\langle \psi_0 | x^* \rangle) = \sin\left(\frac{1}{\sqrt{2^n}}\right) \approx \frac{1}{\sqrt{2^n}}$$

So set $T = \left\lfloor \frac{\pi}{4\theta_0} - \frac{1}{2} \right\rceil \approx \frac{\pi}{4}\sqrt{2^n}$

Analysis

Success probability:

Define: $\hat{T} = \dfrac{\pi}{4\theta_0} - \dfrac{1}{2}$ ➡ $\delta = (T - \hat{T})/2 \in [-1, 1]$

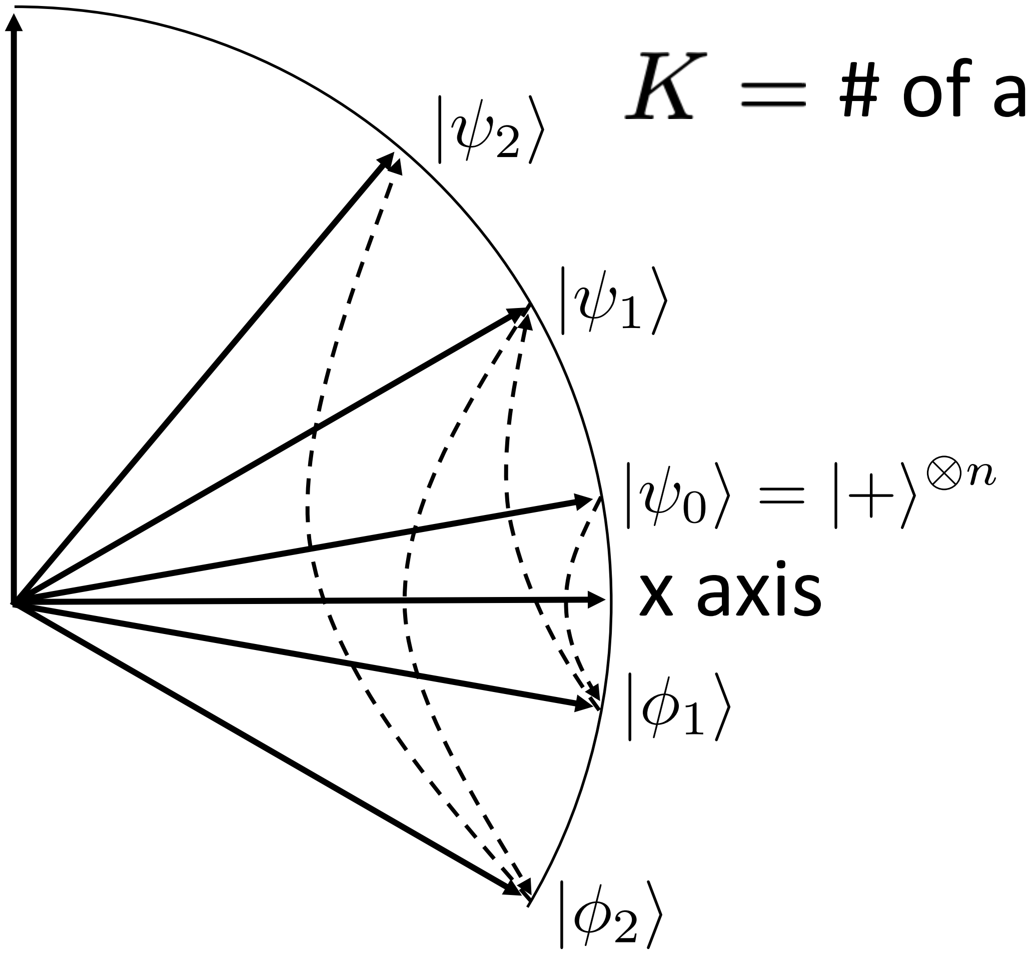
$$\begin{aligned}
|\langle x^* | \psi_T \rangle|^2 &= \sin^2 \left[(2T+1)\theta_0\right] \\
&= \sin^2 \left[(2\hat{T}+1)\theta_0 + \delta\theta_0\right] \\
&= \sin^2 \left[\frac{\pi}{2} + \delta\theta_0\right] \\
&\geq 1 - \delta^2\theta_0^2 \geq 1 - O(2^{-n})
\end{aligned}$$

More generally, can run Grover for any desired number of iterations T, and achieve success probability $O(T^2/2^n)$

What if there is more than one accepting input?

y axis = $|X^*\rangle := \frac{1}{\sqrt{K}} \sum_{x:f(x)=1} |x\rangle$

$K =$ # of accepting inputs



$$\theta_0 = \sin(\langle \psi_0 | X^* \rangle) = \sin\left(\sqrt{\frac{K}{2^n}}\right) \approx \sqrt{\frac{K}{2^n}}$$

Number of evaluations: $\approx \frac{\pi}{4}\sqrt{\frac{2^n}{K}}$

More generally, can run Grover for any desired number of iterations T, and achieve success probability $O(T^2K/2^n)$

Problem: what if don’t know K ?

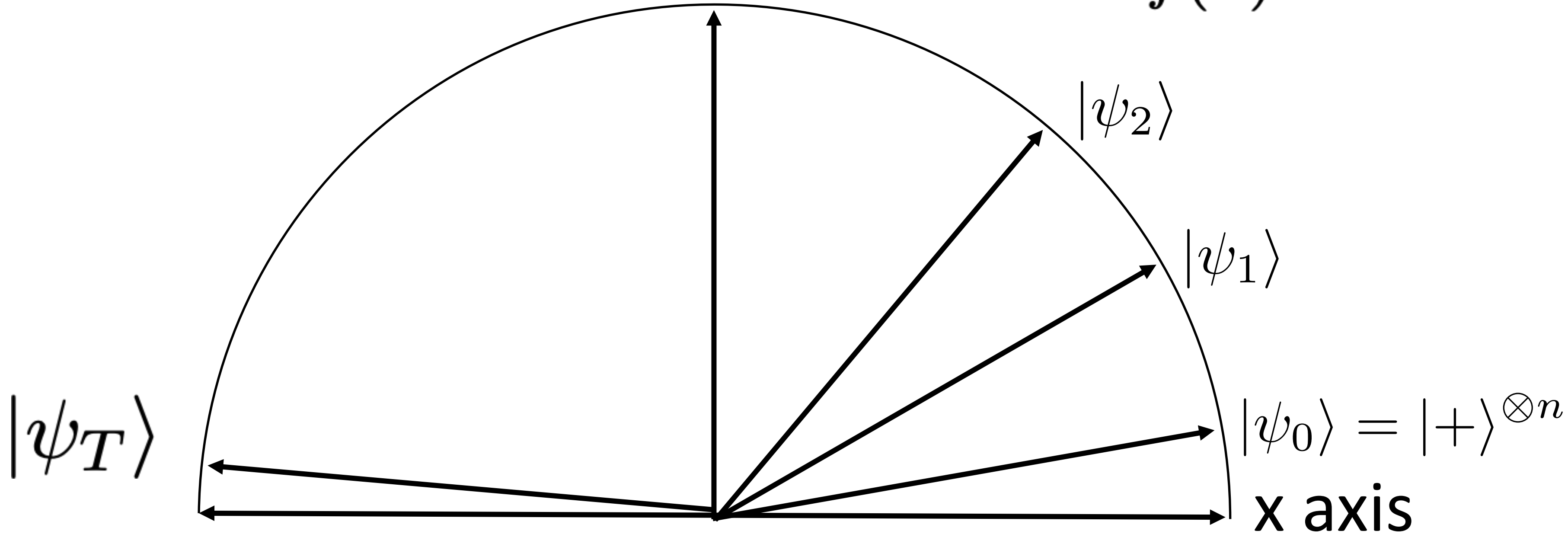
e.g. $K = 4$, but we think its 1

In each step, our angle increases $\approx 2\sqrt{\frac{K}{2^n}} = 4\frac{1}{\sqrt{2^n}}$

We erroneously make $\approx \frac{\pi}{4}\sqrt{2^n}$ steps

Final angle: $\approx \pi$

y axis = $|X^*\rangle := \frac{1}{\sqrt{K}} \sum_{x:f(x)=1} |x\rangle$



Will essentially give random strings

Solution: try powers of 2

```
For $i = 1, 2, \cdots$
    Run Grover's algorithm for $\frac{\pi}{4}\sqrt{2^i}$ steps
    If solution found, stop
```

Will stop roughly when $2^i \approx 2^n/K$

Total iterations: $\frac{\pi}{4}\left(\sqrt{1}+\sqrt{2}+\sqrt{4}+\sqrt{8}+\cdots+\sqrt{\frac{2^n}{K}}\right) \leq O\left(\sqrt{\frac{2^n}{K}}\right)$

Is it possible to do better?

Probably not

In the unlikely event that **NP⊆BQP**, then yes, can do better

However, widely considered unlikely

Some formal evidence that Grover may, in fact, be optimal

Black-box / Query model

Treat f as a black-box, only access is through applications/queries to U_f

Thm: Grover's algorithm is optimal among black-box quantum algorithms

Impact on cryptography

In theory:

$2^{n/2}$ is still exponential time, so does not break any cryptosystem

In practice:

We set parameters so that best attacks take time 2^{128} or 2^{256}

For symmetric key cryptography, best attacks typically were simply brute-force search, so could set key size = 128 or 256

Grover would break such keys in time 2^{64} or 2^{128}

Simple Solution: double key sizes

Keys of length 256 if we want attack time 2^{128}

Keys of length 2^{512} if we want attack time 2^{256}

Downside: crypto will run at half the speed

But actually, maybe not necessary

Observation 1: 2^{100} time is cusp of what is possible today, but only because of extreme parallelism

Any given processor can only perform $\approx 2^{57}$ operations per year

Brute-force search is inherently parallelizeable, but Grover is sequential

Parallel Classical Brute-Force

Total sequential time T

(say, 2^{60})

Total processors P

(say, 2^{40})

Can find answer with probability $\frac{TP}{2^n}$

Can break if $TP \geq 2^n$

Parallel Grover

Total sequential time T

(say, 2^{60})

Total processors P

(say, 2^{40})

Each processor has success probability $\frac{T^2}{2^n}$

Overall success probability $\frac{T^2P}{2^n}$

Can break if $T^2P \geq 2^n$

But actually, maybe not necessary

Observation 2: the overheads for quantum may be huge

Maybe the clock speed is vastly slower than classical

Due to errors, may be very difficult to keep computation going for months

Ultimately, impacts of Grover will depend on how good future quantum hardware is, so hard to tell.

Conservative approach is to double key sizes, but possibly overkill

Next time: Shor’s algorithm

(Exponential quantum speedups)