Lecture 12

Nov 5, 2024

Continuation of prev lecture on Shor's:

Only remains to show with prob. $\geq \frac{1}{32}$,

$$-\frac{r}{2} \leq yr \bmod Q \leq \frac{r}{2}. \qquad (*)$$

With this, we conclude, we generate ord(x) with probability, $\Omega(1/\log N)$.

Pf of $(*)$:

Recall amplitude on $|y\rangle$ is $\frac{1}{\sqrt{QJ}} \omega^{y\ell} \sum_{j=0}^{J-1} \omega^{yrj}$

with $J = \lfloor \frac{Q}{r} \rfloor$

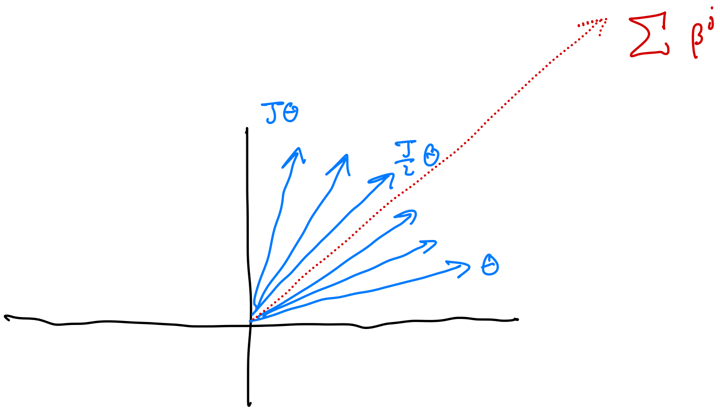
Focus on the $\sum_{j=0}^{J-1} \omega^{yrj}$ term as this is where the constructive/destructive interference occurs. Let $\beta = \omega^{yr} = e^{\left(2\pi i \cdot \frac{yr}{Q}\right)}$

If $\frac{-r}{2} \le yr \bmod Q \le \frac{r}{2}$, then $\beta = e^{i\theta}$ for θ an angle s.t. $|\theta| \le \frac{2\pi r}{2Q} = \pi\left(\frac{r}{Q}\right)$.

Then $\omega^{yrj} = \beta^j$ corresponds to angle $j\theta$ with

$$|j\theta| \le |J\theta| \le \pi.$$

So, the terms of $\sum_{j=0}^{J-1} \omega^{yrj} = \sum_{j=0}^{J-1} \beta^j$ span only angle π.



Simple calculation: $\frac{1}{2}$ the terms make angle $\leq \frac{\pi}{4}$ to resultant vector. Since overall span $\leq \pi$, no vector contributes negatively to resultant vector.

$$\cos\frac{\pi}{4} = \frac{1}{\sqrt{2}}.$$

So, length of resultant vector of $\displaystyle \frac{1}{\sqrt{QJ}}\,\omega^{yl}\sum_{j=0}^{J-1}\omega^{yrj}$

$$\geq \underbrace{\frac{J}{2}}_{\text{half terms}} \cdot \underbrace{\frac{1}{\sqrt{2}}}_{\text{contribution per term}} \cdot \frac{1}{\sqrt{QJ}} = \frac{1}{2^{3/2}} \cdot \frac{1}{\sqrt{\frac{Q}{J}}} \geq \frac{1}{4\sqrt{r}}.$$

Conclusion: If $-\frac{r}{2} \leq yr \bmod Q \leq \frac{r}{2}$, then the resulting vector $|y\rangle$ has magnitude $\geq \frac{1}{4\sqrt{r}}$.

We next show many such vectors y exist.

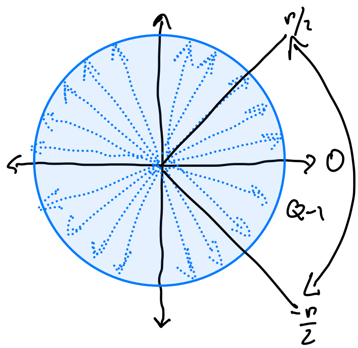
(since $r \in \mathbb{Z}_Q^*$).

If $\gcd(r, Q) = 1$, then $\exists\, r^{-1}$ s.t. $r \cdot r^{-1} \equiv 1 \mod Q$.

Therefore the map $y \longmapsto yr$ is a permutation of $\{0 \ldots Q-1\}$.

So, at least r vectors y exist. s.t. $-\frac{r}{2} \leq yr \mod Q \leq \frac{r}{2}$.

Cartoon:



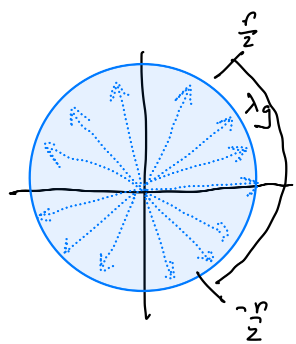
all possible values occur as yr.

So r vectors in region.

When $g := \gcd(r, Q) > 1$, note $g \leq r/2$. Then yr are uniformly distributed over

$$0,\ g,\ 2g,\ \cdots,\ \left(\frac{Q}{g} - 1\right) g$$

with $yr = \lambda g$ for g values of y.



If $|\lambda g| \le \frac{r}{2} \Rightarrow |\lambda| \le \frac{r}{2g}$

Then, for at least $2\left\lfloor \frac{r}{2g} \right\rfloor \cdot g \ge \frac{r}{2}$ vectors y,

$-\frac{r}{2} \le yr \bmod Q \le \frac{r}{2}$.

So, total probability mass on y s.t. $-\frac{r}{2} \le yr \bmod Q \le \frac{r}{2}$.

is $\ge \frac{r}{2} \cdot \left(\frac{1}{4\sqrt{r}}\right)^2 = \frac{1}{32}$.

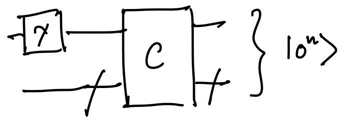
Therefore, we sample a y according to the algorithm for order finding, we correctly calculate $\mathrm{ord}(x)$ with probability $\Omega\left(\frac{1}{\log N}\right)$.

So our overall algorithm for factoring is efficient in that it runs in time $\mathrm{polylog}(N)$.

Next time: efficient classical algorithms for simulating quantum computation.

Today: Efficient classical algorithms for simulating quantum computations

Problem: given an input $\langle C \rangle$ the description of a q. circuit with n qubits, m gates, and no measurements, what is the probability that



the measurement output is 1?

ω = mat. mult. coefficient.

Computing $p = \| \langle 1 | \otimes \mathbb{1} | C | 0^n \rangle \|^2$ to accuracy ϵ can be solved in classical time $O(2^{\omega n} \log(\frac{m}{\epsilon}))$ and space $O(2^n \log(\frac{m}{\epsilon}))$.

Pf. Let $C = g_m g_{m-1} \cdots g_1$.

For gate g_t, let $\tilde{g}_t$ be the pruning of g_t to ℓ bits.

Then $\| \tilde{g}_t - g_t \|_F = \sqrt{\sum_{ij} (\tilde{g}_t - g_t)_{ij}^2} = 4 \cdot 2^{-\ell}$ for 4×4 matrices.

Since $\| \cdot \| \le \| \cdot \|_F$,

$$\|\tilde{g}_t - g_t\| \leq 4 \cdot 2^{-\ell} \quad \text{so} \quad \|\tilde{g}_t \otimes \mathbb{1} - g_t \otimes \mathbb{1}\| \leq 4 \cdot 2^{-\ell}.$$

Then for $\tilde{C} = \tilde{g}_m \tilde{g}_{m-1} \cdots \tilde{g}_1$,

$$\|\tilde{C}|0^n\rangle - C|0^n\rangle\| \leq 4m \cdot 2^{-\ell}$$

So $|\tilde{p} - p| \leq 8m \cdot 2^{-\ell} \leq \epsilon$

Choose ℓ s.t. $8m \cdot 2^{-\ell} \leq \epsilon \Rightarrow \ell \geq \Omega\left(\log \frac{m}{\epsilon}\right)$.

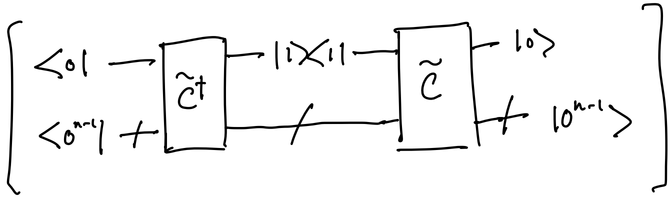
Compute $\tilde{p}$ using mat. multiplication. $|p - \tilde{p}| \leq \epsilon$.

Additionally, we can multiply and prune as we compute.

gives a runtime of $O\Big(\underset{\substack{\uparrow \\ \text{mat. mult.}}}{2^{\omega n}} \underset{\substack{\uparrow \\ \text{size of integers}}}{\log\left(\frac{m}{\epsilon}\right)}\Big)$ and space $O\left(2^n \log\left(\frac{m}{\epsilon}\right)\right)$.

In actuality, no one uses such fast matt. algorithms since the coefficients are huge. So runtime is more like $O\left(2^{2.87n} \log^2\left(\frac{m}{\epsilon}\right)\right)$.

<u>Claim</u> We can reduce the space complexity to $\text{poly}\left(n, \log\left(\frac{1}{\epsilon}\right)\right)$.

<u>Pf.</u> $\tilde{p} =$ 

$$\tilde{p} = \langle 0^n | \tilde{g}_1^\dagger \tilde{g}_2^\dagger \cdots \tilde{g}_m^\dagger \left(|1\rangle\langle 1| \otimes \mathbb{1} \right) \tilde{g}_m \cdots \tilde{g}_1 |0^n\rangle$$

(one big matrix multiplication)

Add identity terms $\mathbb{1} = \sum_{y \in \{0,1\}^n} |y\rangle\langle y|$.

$$\tilde{p} = \langle 0^n | \tilde{g}_1 \Big(\sum_{y_{2m+1}} |y_{2m+1}\rangle\langle y_{2m+1}| \Big) \tilde{g}_2 \Big(\sum_{y_{2m}} |y_{2m}\rangle\langle y_{2m}| \Big) \cdots \Big(\sum_{y_1} |y_1\rangle\langle y_1| \Big) \tilde{g}_1 |0^n\rangle$$

$$= \sum_{y_1, \ldots, y_{2m+1}} \langle 0^n | \tilde{g}_1 | y_{2m+1} \rangle \cdots \langle y_1 | \tilde{g}_1 | 0^n \rangle .$$

Alg: Iterate over $y_1, \ldots, y_{2m+1} \in \{0,1\}^n$ computing each multiplication in the sum. Requires

$$O\left(2^{2nm} \log^2\left(\tfrac{m}{\epsilon}\right)\right) \text{ time but only } O\left(nm + \log\left(\tfrac{m}{\epsilon}\right)\right) \text{ space.}$$

To estimate p to $1/6$, requires only $O(nm)$ space.

Proves $BQP \subseteq PSPACE$. (Called the Feynman path integral)

i.e. every q. computation can be simulated with polynomial space but (perhaps) exponential time.

Next: A situation when we can vastly improve the time complexity.

The issue is that keeping track of the state $g_t \cdots g_1 |0^n\rangle$ is inefficient and may take 2^n complex numbers to record.

One solution was to keep "no numbers" using path integral.

Another is succinct descriptions of q. states.

First, Pauli matrices:

$$\mathbb{1}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = iXZ, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

X, Y, Z all anticommute, have trace 0, square to $\mathbb{1}$.

$$\mathcal{P}_1 = \{\pm \mathbb{1}, \pm i\mathbb{1}, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}.$$

a group under matrix multiplication.

$$\mathcal{P}_n = \{P_1 \otimes P_2 \otimes \cdots \otimes P_n \mid P_1, \ldots, P_n \in \mathcal{P}_1\}.$$ Also a group.

Pauli matrices can be described with $2(n+1)$ bits.

Use notation: X_j to denote $\mathbb{1} \otimes \dots \otimes \mathbb{1} \otimes \underset{\substack{\uparrow \\ j^{\text{th}} \text{ location.}}}{X} \otimes \mathbb{1} \otimes \dots \otimes \mathbb{1}$

So
$$
\begin{aligned}
(X_1 Z_2)(X_2 Y_3) &= (X \otimes Z \otimes \mathbb{1})(\mathbb{1} \otimes X \otimes Y) \\
&= X \otimes ZX \otimes Y \\
&= X \otimes iY \otimes Y = i(X \otimes Y \otimes Y) \\
&= i\, X_1 Y_2 Y_3 .
\end{aligned}
$$

An observation: $|0^n\rangle$ is the unique solution to $Z_j|\psi\rangle = |\psi\rangle$, for all $j = 1, \dots, n$.

<u>Pf</u>. $Z_1|\psi\rangle = |\psi\rangle$ only if $|\psi\rangle = |0\rangle \otimes |\psi'\rangle$.
Rest follows similarly. $\square$

Another observation: $|+\rangle^{\otimes n}$ is the unique solution to $X_j|\psi\rangle = |\psi\rangle$, for all $j = 1, \dots, n$.

<u>Lemma</u> Assume $|\psi\rangle$ is the unique solution to $P_j|\psi\rangle = |\psi\rangle$ for Pauli matrices $P_1, \dots, P_n$. Let $\mathcal{U}$ be any unitary.

Define $Q_j = \mathcal{U} P_j \mathcal{U}^\dagger$. Then $\mathcal{U}|\psi\rangle$ is the unique solution to $Q_j|\tau\rangle = |\tau\rangle$ for all $j = 1, \dots, n$.

Pf. To see it is a solution, notice

$$\begin{aligned} Q_j \mathcal{U}|\psi\rangle &= \mathcal{U} P_j \mathcal{U}^\dagger \mathcal{U} |\psi\rangle \\ &= \mathcal{U} P_j |\psi\rangle \\ &= \mathcal{U}|\psi\rangle. \end{aligned}$$

For uniqueness, assume $\exists$ a solution $|\tau\rangle$. Then,

$$|\tau\rangle = Q_j|\tau\rangle \implies \mathcal{U}^\dagger|\tau\rangle = P_j \mathcal{U}^\dagger|\tau\rangle \quad \forall j = 1, \dots, n.$$

So, $\mathcal{U}^\dagger|\tau\rangle = |\psi\rangle$ by uniqueness. So $|\tau\rangle = \mathcal{U}|\psi\rangle$. $\square$

If $|\psi\rangle$ is the unique state s.t. $P_j|\psi\rangle = |\psi\rangle$ for all $j = 1, \dots, n$, we say $P_1, \dots, P_n$ <u>stabilize</u> $|\psi\rangle$.

Issue is that for arbitrary $\mathcal{U}$, $\mathcal{U} P_j \mathcal{U}^\dagger$ may not be a Pauli matrix.

But for some $\mathcal{U}$ it will be. The set of $\mathcal{U}$ for which $\mathcal{U} P \mathcal{U}^\dagger$ is also a Pauli matrix $\forall P$ is called the normalizer group of $\mathcal{P}_n$. The normalizer group of $\mathcal{P}_n$ is called the Clifford group, C_n.

$$C_n = \{ \mathcal{U} \mid \mathcal{U} P \mathcal{U}^\dagger \in \mathcal{P}_n \ \forall \ P \in \mathcal{P}_n \}.$$

It's a more complicated pf than we have time for this class, but every matrix $\in C_n$ can be generated from

$CNOT \otimes \mathbb{1}_{n-2}$, $S \otimes \mathbb{1}_{n-1}$, $H \otimes \mathbb{1}_{n-1}$. and their $\pm$, $\pm i$ variants.

Here, $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$.

Many other unitaries such as X, Y, Z, SWAP, CZ are all part of the Clifford group.

Consider H_1 in the Clifford group C_n.

Suppose $P_1 \ldots P_n$ stabilize $|\psi\rangle$.

Then, we can efficiently calculate stabilizers for $H|\psi\rangle$.

If $P_j = \mathbb{1} \otimes \_\_\_\_$, then $H_1 P_j H_1^\dagger = \mathbb{1} \otimes \_\_\_\_ = P_j$.

If $P_j = X \otimes \_\_\_\_$, then $H_1 P_j H_1^\dagger = Z \otimes \_\_\_\_$

If $P_j = Z \otimes \_\_\_\_$, then $H_1 P_j H_1^\dagger = X \otimes \_\_\_\_$

If $P_j = Y \otimes \_\_\_\_$, then $H_1 P_j H_1^\dagger = Y \otimes \_\_\_\_$

Similar rules can be generated for CNOT and S updates.

Gottesman-Knill

Thm given a circuit $g_m \cdots g_1$ with each $g_t \in \{CNOT, S, H\}$, we can efficiently compute a collection of stabilizers for

$g_m \cdots g_1 |0^n\rangle$.

Pf. Starting with $P_j = Z_j$ which stabilize $|0^n\rangle$, we update stabilizers gate by gate. Each update takes $O(n^2)$ time as there are n stabilizers each of $O(n)$ bits. Total time is $O(mn^2)$, space $O(n^2)$. □

What about measurements?

Wlog, we only need to consider measuring the first qubit in standard basis.

Notice if $P|\psi\rangle = P'|\psi\rangle = |\psi\rangle$ for Paulis P, P', then

$PP'|\psi\rangle = P|\psi\rangle = |\psi\rangle$ so PP' stabilizes $|\psi\rangle$ as well.

So if $P_1, \dots, P_n$ stabilize $|\psi\rangle$ then

$\langle P_1, \dots, P_n \rangle$ stabilizes $|\psi\rangle$ where this is the stabilizer

subgroup $\leq \mathcal{P}_n$.

Let $S_\psi = \{ P \in \mathcal{P}_n \mid P|\psi\rangle = |\psi\rangle \}$.

Measuring $|\psi\rangle$:

1. If $Z_1 \in S_\psi$, then measurement outcome is 0 and state doesn't change. Deterministic measurement.

2. If $-Z_1 \in S_\psi$, then measurement outcome is 1 and state doesn't change. Deterministic measurement.

3. If $Z_1 \notin S_\psi$, things get more complicated.

   Z_1 must not commute with all of S_ψ.

   Find a basis for S_ψ s.t. $S_\psi = \langle b_1, \dots, b_n \rangle$, and $b_1 Z_1 = -Z_1 b_1$ but $b_j Z_1 = Z_1 b_j$ for $j > 1$.

Flip a coin. Replace b_1 with Z_1 or $-Z_1$ depending on the coin flip.

Pf of correctness

Since b_1 and Z_1 anticommute, square to $\mathbb{1}$, by prev 2 problem, there exists a change of basis s.t.
$\mathcal{U} b_1 \mathcal{U}^\dagger = X_1$ and $\mathcal{U} Z_1 \mathcal{U}^\dagger = Z_1$, and $\mathcal{U} b_j \mathcal{U}^\dagger = \mathbb{1} \otimes b_j'$.

Since $b_1 \in S_\psi$, $\mathcal{U}|\psi\rangle = |+\rangle \otimes$ ____

So measuring, Z_1 is a coin-flip resulting in $|0\rangle$ or $|1\rangle$. Doesn't change remainder of state, so new state is stabilized by $Z_1, b_2, \ldots, b_n$ or $-Z_1, b_2, \ldots, b_n$ depending on outcome. $\square$

Finding a basis $\langle b_1, \ldots b_n \rangle$ for S_ψ s.t. only b_1 anticommutes:

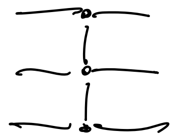
1. Renumber bases s.t. b_1 anticommutes.

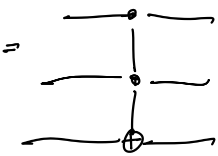
② If b_k anticommutes, replace b_k with $b_1 b_k$.

Next, computation with a few non-Clifford gates.

non-Clifford gate examples:

$$T = \sqrt{S} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{pmatrix}$$

CCZ 

CCX = 

(Toffoli)

<u>Theorem</u> (Solovay-Kitaev) Any 2-qubit unitary can be ϵ-approximated using $O(\text{polylog}(1/\epsilon))$ H, T, CNOT gates.

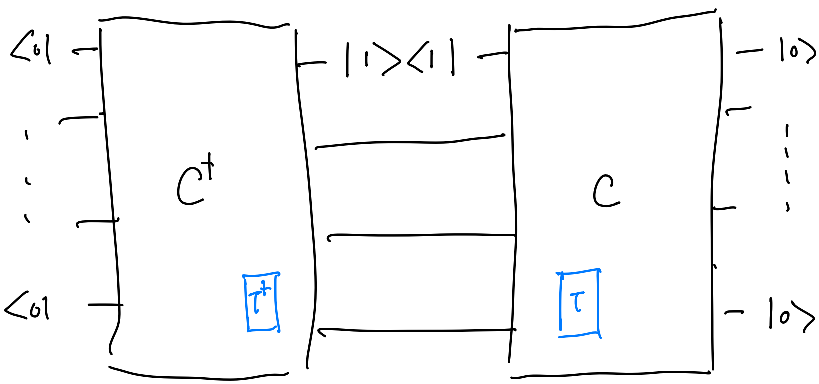
Solovay-Kitaev + Clifford simulation suggests that the number of T gates in a H, T, CNOT circuit should be a measure of the circuits complexity.

Thm $\exists$ a constant $\alpha > 0$, s.t. computing the output probability of a quantum circuit consisting of m- Clifford gates, t T-gates on n qubits can be classically computed in time $O(2^{\alpha t} \cdot \mathrm{poly}(n,m))$.

Best: $\alpha < 0.4$ (Qassim-Pashayan-Gosset)

Today $2^\alpha = 3$, $\alpha < 1.6$.

Model of such a computation:



$\uparrow$ one big matrix multiplication:

Replacement :

$$T^\dagger \otimes T = a\, \mathbb{1} \otimes \mathbb{1} + b\, S^\dagger \otimes S + c\, Z^\dagger \otimes Z.$$

$$\begin{pmatrix} 1 & & & \\ & e^{i\pi/4} & & \\ & & e^{-i\pi/4} & \\ & & & 1 \end{pmatrix} = \begin{pmatrix} a & & & \\ & a & & \\ & & a & \\ & & & a \end{pmatrix} + \begin{pmatrix} b & & & \\ & bi & & \\ & & -bi & \\ & & & b \end{pmatrix} + \begin{pmatrix} c & & & \\ & -c & & \\ & & -c & \\ & & & c \end{pmatrix}$$

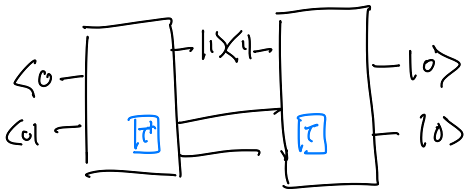
Solve

$$a + b + c = 1 \qquad a = 1/2$$

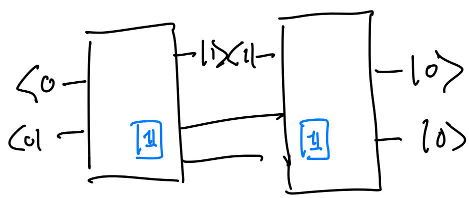
$$a + bi - c = e^{i\pi/4} \qquad b = 1/\sqrt{2}$$

$$a - bi - c = e^{-i\pi/4}. \qquad c = \frac{1}{2} - \frac{1}{\sqrt{2}}.$$

By linearity,

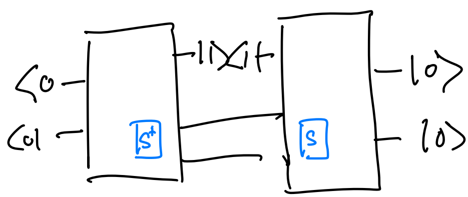


= a



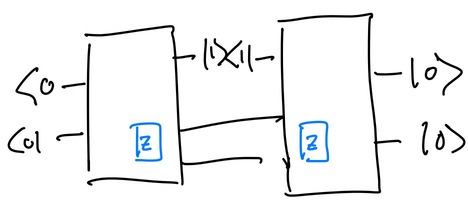
+

b



+

c



Apply this replacement recursively for every pair of T gates.

Yields 3^t calculations each of which was a only Clifford computation. So previous, subroutine gives an efficient poly(n,m) algorithm.