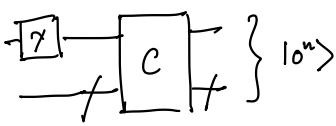
Lecture 14

Nov 10, 2025

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 = matr. mult. coefficient

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

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 $\| \cdot \| \leq \| \cdot \|_F$,

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

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.

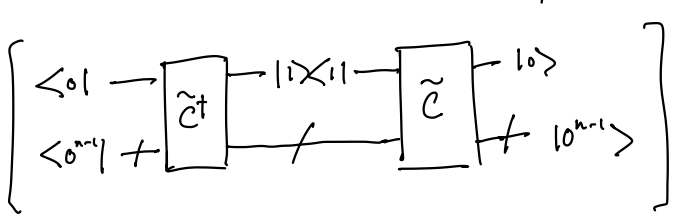
gives a runtime of $O\left(2^{\omega n} \log\left(\frac{m}{\epsilon}\right)\right)$ and space $O\left(2^n \log\left(\frac{m}{\epsilon}\right)\right)$.

(ω: mat. mult.; $\log\left(\frac{m}{\epsilon}\right)$: size of integers)

In actuality, no one uses such fast matt. algorithms since the coefficients are huge. So runtime is more like $O\left(2^{2.77n} \log^2\left(\frac{m}{\epsilon}\right)\right)$.

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

Pf.



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

(one big matrix multiplication)

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

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

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

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

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

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 $g_t \dots g_1 |0^n\rangle$ 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 $\mathbb{1}$.

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

a group under matrix multiplication.

$$\mathcal{P}_n = \left\{ P_1 \otimes P_2 \otimes \dots \otimes P_n \;\middle|\; P_1, \dots, P_n \in \mathcal{P}_1 \right\}.$$ Also a group.

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

Use notation: X_j to denote $\mathbb{1} \otimes \cdots \otimes \mathbb{1} \otimes X \otimes \mathbb{1} \otimes \cdots \otimes \mathbb{1}$

(X in the j^{th} location.)

So $(X_1 Z_2)(X_2 Y_3) = (X \otimes Z \otimes \mathbb{1})(\mathbb{1} \otimes X \otimes Y)$

$= X \otimes ZX \otimes Y$

$= X \otimes iY \otimes Y = i(X \otimes Y \otimes Y)$

$= i\, X_1 Y_2 Y_3$.

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

Pf. $Z_1 |\psi\rangle = |\psi\rangle$ only if $|\psi\rangle = |0\rangle \otimes |\psi'\rangle$.

Rest follows similarly. □

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

Lemma Assume $|\psi\rangle$ is the unique solution to $P_j |\psi\rangle = |\psi\rangle$ for Pauli matrices $P_1, \ldots, P_n$. Let U be any unitary.

Define $Q_j = UP_jU^\dagger$. Then $U|\psi\rangle$ 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

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

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

$$|\tau\rangle = Q_j|\tau\rangle \implies U^\dagger|\tau\rangle = P_jU^\dagger|\tau\rangle \quad \forall j = 1, \dots, n.$$

So, $U^\dagger|\tau\rangle = |\psi\rangle$ by uniqueness. So $|\tau\rangle = U|\psi\rangle$. ☐

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

Issue is that for arbitrary U, $UP_jU^\dagger$ may not be a Pauli matrix.

But for some U it will be. The set of U for which $UPU^\dagger$ 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.

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

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, in the Clifford group C_n.

Suppose $P_1 \cdots P_n$ 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 $g_\ell \in \{\text{CNOT}, S, H\}$, 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.

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 $g_\ell \in \{\text{CNOT}, S, H\}$, 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 $P_1, \ldots, P_n$ stabilize $|\psi\rangle$ then

$\langle P_1, \ldots, P_n \rangle$ 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. $S_\psi = \langle b_1, \ldots, b_n \rangle$, 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 prob 2 problem, there exists a change of basis s.t.

$U b_1 U^\dagger = X_1$ and $U Z_1 U^\dagger = Z_1$, and $U b_j U^\dagger = \mathbb{1} \otimes b_j'$.

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

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. □

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

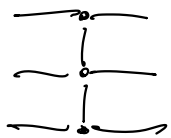
① Renumber bases s.t. b_1 anticommutes.

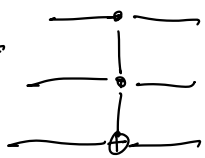
(2) If b_k anticommutes, replace b_k with $b_1 b_k$.

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)

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

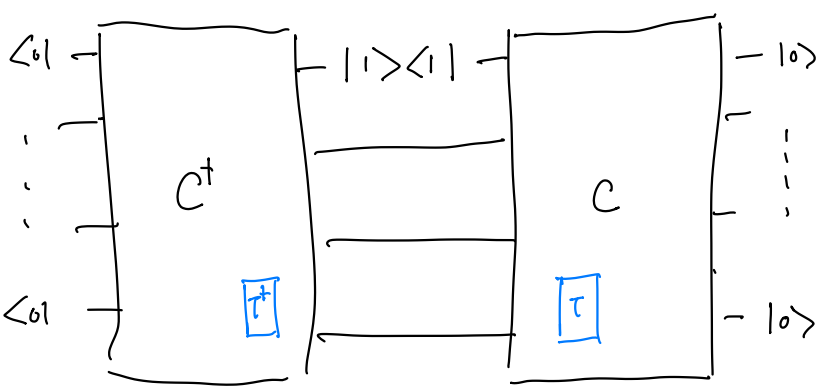
Solovay-Kitaev + Clifford simulation suggests that the number of T gates in a H, T, CNOT circuit should be a measure of the circuits complexity.

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 $O(2^{\alpha t} \cdot \text{poly}(n, m))$.

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

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

Model of such a computation:



↑ one big matrix multiplication:

Replacement :

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

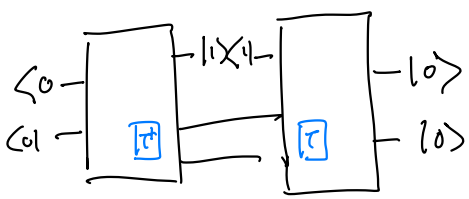
$$\begin{pmatrix} 1 & & & \\ & e^{i\pi/4} & & \\ & & e^{-i\pi/4} & \\ & & & 1 \end{pmatrix} = \begin{pmatrix} a & & & \\ & a & & \\ & & a & \\ & & & a \end{pmatrix} + \begin{pmatrix} b & & & \\ & bi & & \\ & & -bi & \\ & & & b \end{pmatrix} + \begin{pmatrix} c & & & \\ & -c & & \\ & & -c & \\ & & & c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

Solve $a+b+c=1$ $\quad a = 1/2$

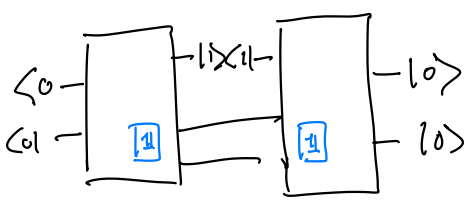
$a+bi-c = e^{i\pi/4}$ $\quad b = 1/\sqrt{2}$

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

By linearity,

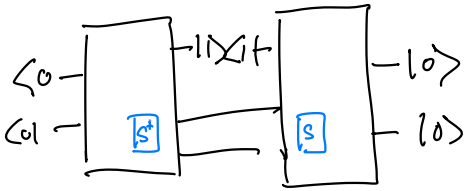


$= a$



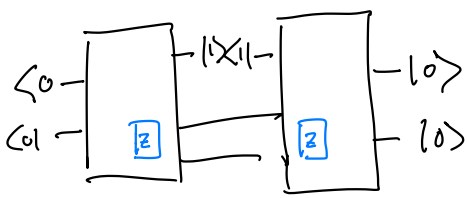
$+$

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.

Quantum Complexity Theory

we define BQP - the class of decision problems decidable in poly-time by a family of uniform quantum circuits.

Other complexity classes.

P - decision problems solvable by deterministic classical polynomial time computation

BPP - decision problems solvable by randomized classical polynomial time computation

NP – decision problems solvable by non-deterministic classical polynomial time computation

also known as efficiently verifiable decision problems.

interaction-perspective:

Prover

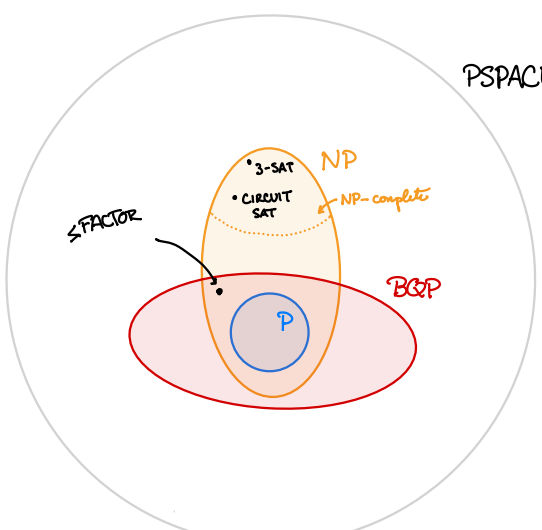
$\downarrow \pi \in \{0,1\}^{\text{poly}(n)}$

Verifier

$V(x, \pi)$ poly-time computation.

$x \in \mathcal{L}$ if $\exists\ \pi$ s.t. $V(x,\pi)$ accepts

$x \notin \mathcal{L}$ if $\forall\ \pi$, $V(x,\pi)$ rejects.



$$\leq \text{FACTOR} = \{ \underbrace{(N, K)}_{\text{as binary numbers}} : N \text{ has a factor} \leq K \}.$$

Useful to understand the notion of reductions.

Def. Promise lang $\mathcal{L}$ poly-time reduces to $\mathcal{L}'$ if

$\exists$ a poly-time algorithm $f : \{0,1\}^* \to \{0,1\}^*$ s.t.

(1) $x \in \mathcal{L}_{yes}$ iff $f(x) \in \mathcal{L}'_{yes}$

(2) $x \in \mathcal{L}_{no}$ iff $f(x) \in \mathcal{L}'_{no}$.

Not. $\mathcal{L} \leq \mathcal{L}'$.

"if we can solve $\mathcal{L}'$, then we can solve $\mathcal{L}$". $\mathcal{L}'$ is as hard as $\mathcal{L}$.

Not. A lang $\mathcal{L}'$ is hard for a comp. class C if

$\forall\ \mathcal{L} \in C,\ \mathcal{L} \leq \mathcal{L}'$.

$\mathcal{L}'$ is C-complete if $\mathcal{L}' \in C$ and $\mathcal{L}'$ is C-hard.

Ex. 3-SAT is NP-complete
Traveling Salesmen is NP-complete.

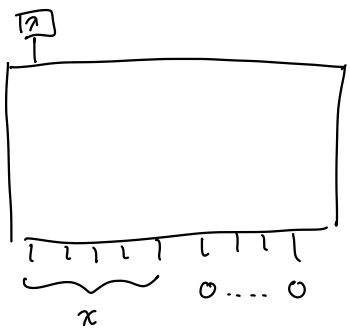
Ex. $(\leq_{\text{FACTOR}}) \leq (\leq_{\text{ORDER-FINDING}})$

Circuit-sat is NP-complete.

Input: $\langle C \rangle$ ← classical boolean reversible circuit

with some free wires and some fixed ancilla.

Decide if $\exists\, x$ s.t. $C(x)$ accepts.



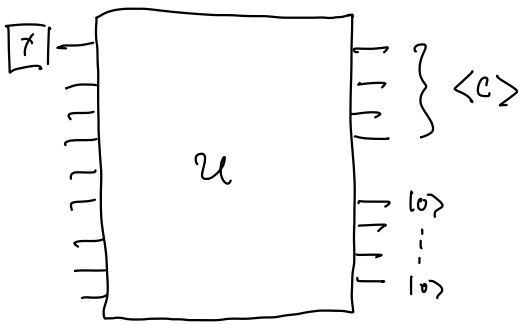
BQP-complete: Input $\langle C \rangle$ ← quantum circuit

Decide if ① yes: $\left\| \langle 1 | \otimes \mathbb{1} | C | 0^n \rangle \right\|^2 \geq \frac{2}{3}$

② no: _____ $\leq 1/3$.

"Canonical BQP-complete problem".

Containment $\in$ BQP is because of the notion of a universal quantum circuit.



s.t. pr measurement = 1 is equal to success prob. of C.

$\Rightarrow$ BQP $\subseteq$ PSPACE as we gave a PSPACE alg for this problem.

$$\mathcal{L}_{a,b} = \left\{ \langle C \rangle \text{ s.t. } p_C \geq a \quad (\text{yes}) \text{ or } p_C \leq b \quad (\text{no}) \right\}$$

where $\| \langle 1 | C | 0^n \rangle \|^2 =: p_C$.

Claim $\mathcal{L}_{a,b} \in$ BQP for $\epsilon := a - b \geq$

Pf. Use Universal q.c. to run Circuit C T times getting outcomes $X_1, \ldots, X_T$. wlog assume $a \geq \frac{1}{2}$.

if $\langle C \rangle \in \mathcal{L}_{yes}$ (i.e. $p_C \geq a$), then

$\mathbb{E} X_t \geq a$. So $X = \sum X_t$.

$$\Pr\left[\mathbb{E}X \le b\right] = \Pr\left[X \le (aT)-(\epsilon T)\right]$$

$$= \Pr\left[aT - X \ge \epsilon T\right]$$

$$\le \Pr\left[aT - X \ge \epsilon aT\right]$$

$$\le \exp\left(-\frac{\epsilon^2 aT}{3}\right) \le \exp\left(-\frac{\epsilon^2 T}{6}\right)$$

only need error bound of $\le \frac{1}{3}$.

So $T \ge \Omega\left(\frac{1}{\epsilon^2}\right)$.

BQP can estimate p_c to accuracy $1/\text{poly}(n)$.

we can also boost success probability to $2^{-p(n)}$ of outputting correct answer by choosing $T \ge \Omega\left(\frac{p(n)}{\epsilon^2}\right)$.