4.1 Quantum Algorithms

- What's quantum good for?
- Many important problems are unsolvable classically because of the resources they require
- Spectacular promise of quantum is to enable new algorithms for these problems that are feasible.

2 Main Classes of Algos:

① Shor's quantum fourier transform

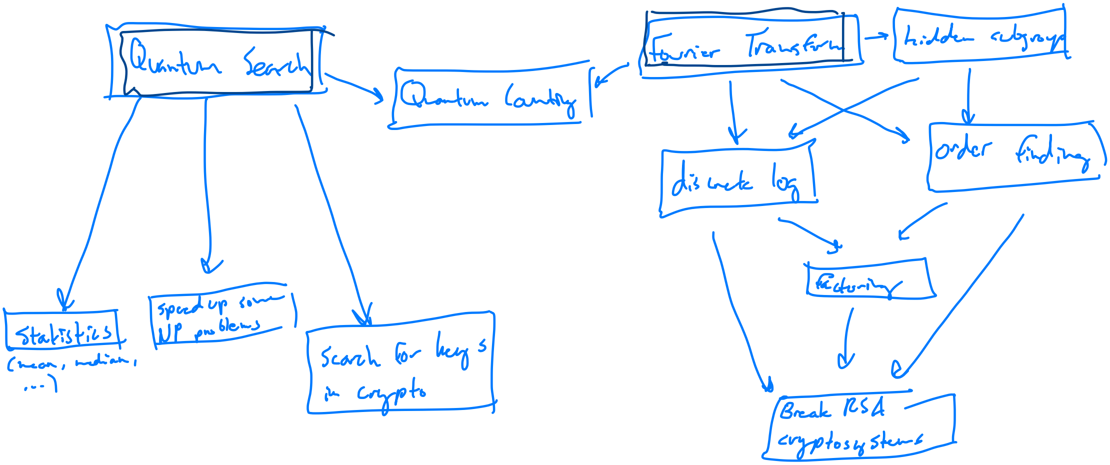
- Exponential speedups from best known classical algos
- Solves factoring & discrete logarithm problems.
  - Break RSA encryption
  - Closely related to finding a hidden subgroup (Generalization of finding the period of periodic function

② Grover's quantum Search

- Quadratic speedup over classical
- Searching is everywhere! - Speed up lots of algos!
- Extract statistics on unordered set (min, mean,...)
- Speed up some algos in NP requiring search
- Speed up search for keys in cryptosystems such as DES.

Quantum Counting:

- Clever combo
- estimate # of solutions to a search problem



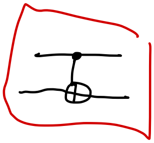
Why so few?

- requirement already that quantum algos are better than classical
- requires special insights and tricks

4.3 Controlled Operations

"If A is true then do B"

- How can complex controlled operations be built from quantum circuits using elementary operations?
- Recall CNOT
  - If control qubit is $|1\rangle$, target is flipped, Otherwise do nothing.



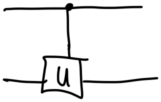
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- More general:

Let U an arbitrary single qubit unitary operation.

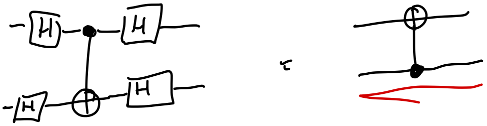
Controlled-U is the same as CNOT but with U.

i.e. $|c\rangle|t\rangle \to |c\rangle \underset{|1\rangle}{\boxed{U^c}}|t\rangle$



- Unlike ideal classical gates, ideal quantum gates don't have 'high impedance' inputs.
- Role of control & target are arbitrary.

Eg



In this basis, the target & control have interchanged roles.

where

$|+\rangle|+\rangle \to |+\rangle|+\rangle$

$|-\rangle|+\rangle \to |-\rangle|+\rangle$

$|+\rangle|-\rangle \to |-\rangle|-\rangle$

$|-\rangle|-\rangle \to |+\rangle|-\rangle$

with basis

$|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$

How to implement controlled-U for arbitrary unitary U?

From Cor. 4.2 on pg. 176,

$\boxed{U = e^{i\alpha} A \boxed{X} B \boxed{X} C}$, where $\boxed{ABC = I}$

- First step is to phase shift exp(iα) on target qubit, controlled by control qubit

  $|00\rangle \rightarrow |00\rangle$, $|01\rangle \rightarrow |01\rangle$, $|10\rangle \rightarrow e^{i\alpha}|10\rangle$, $|11\rangle \rightarrow e^{i\alpha}|11\rangle$

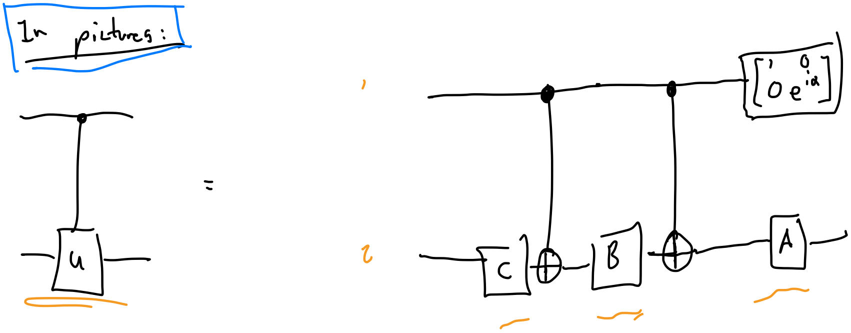
- Suppose control qubit is set:

  then $e^{i\alpha} AXBXC = U$ is applied to 2nd qubit.

- 'spose not: $|0\rangle$

  then $ABC = I$ applied to 2nd qubit.

In pictures:

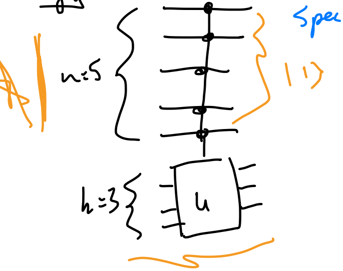


What about conditioning on multiple qubits?

'Spose we have n+k qubits and U is a k-qubit unitary operator. Then

$$\boxed{C^n(U)|x_1 \dots x_n\rangle|\psi\rangle = |x_1 \dots x_n\rangle U^{x_1 x_2 \dots x_n}|\psi\rangle}$$

E.g.)

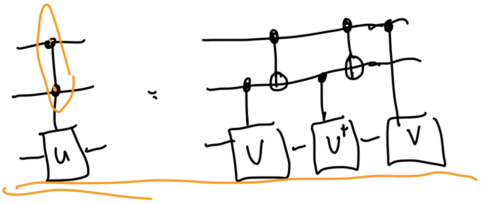


Special notation

But, for k ≥ 2, don't yet know how to perform arbitrary operations on k qubits. Above trick won't work.

2 control qubit operations:

Suppose U is a single qubit unitary operator, and V is a unitary operator s.t. $V^2 = U$. Then $C^2(U)$ can be implemented by



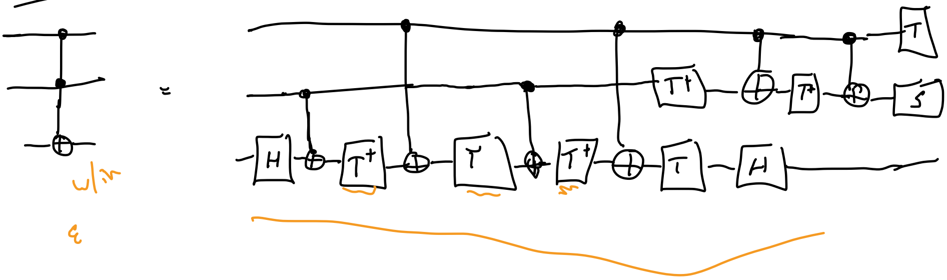
(Special case where $U \equiv (1-i)(I+iX)/2$ is the Toffoli gate)

(Indeed, $U^2 = X$, one may verify the above gate is Toffoli gate)

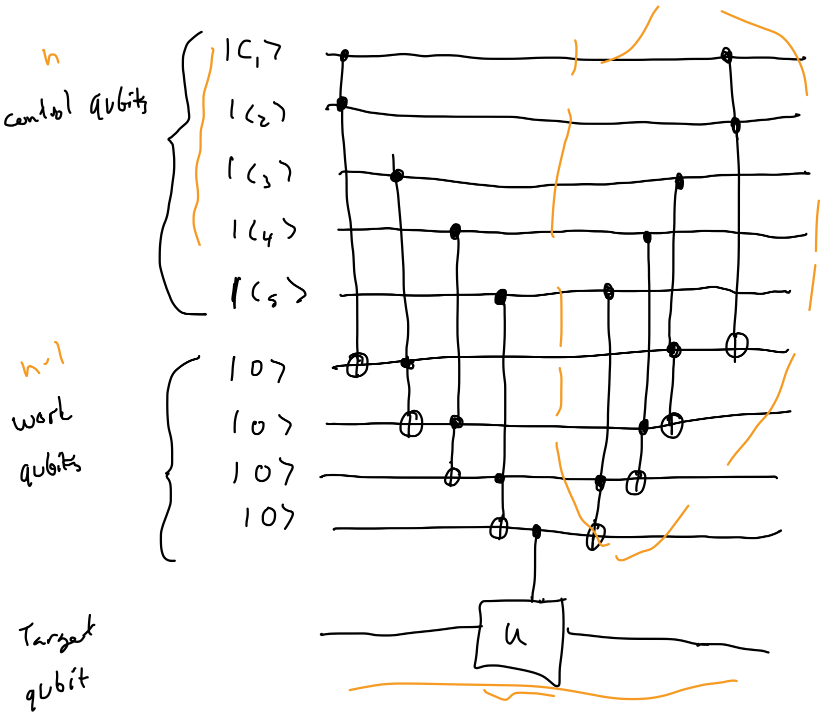
In a classical context, this is remarkable.

- 1 and 2 bit reversible gates aren't enough to implement Toffoli gate in a classical setting, much less universal computation
- Ultimately, any unitary operation can be composed into an arbitrarily good approximation from just Hadamard, phase, controlled-NOT, and π/8 gates.

Eg with Toffoli gate:



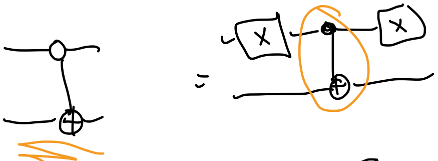
How to implement $C^n(U)$ gates? Simple circuit to do so is as follows:



produces product $|c_1 \cdot c_2 \cdot c_3 \cdot c_4 \cdot c_5\rangle$, and U is then applied if product is 1.

Then work qubits switched back to zero.

These examples have been conducting conditional dynamics if control bits are set to one. But there is nothing special about ones



c-NOT w/ zero conditional.