Portland State U

F, 04/24/2020

S′20 CS 410/510

Intro to quantum computing

Fang Song

Week 4

- Simon's algorithm
- Reversible computation

Credit: based on slides by Richard Cleve

Exercise: Hadamard

1. What is $H^2 := HH$?

$$= I$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad H^{\dagger} = H$$

DEF. of Unitary

$$H^{\dagger}H = I$$

2. What is the matrix form of $H^{\otimes 2} := H \otimes H$?

$$H^{\otimes 2} = H \otimes H = \frac{1}{2}\begin{pmatrix} & & & \\ & & & \end{pmatrix}_{4\times 4}$$

3. Let $|\psi\rangle = \frac{1}{\sqrt{2^3}}\sum_{x\in\{0,1\}^3}|x\rangle$. What is $H^{\otimes 3}|\psi\rangle$?

$$\boxed{H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}}\sum_{y\in\{0,1\}^n}(-1)^{x\cdot y}|y\rangle}$$

$$x\cdot y = x_1y_1 + x_2y_2 + \cdots + x_ny_n \mod 2$$

$$|\psi\rangle = \frac{1}{\sqrt{2^3}}\sum_{x\in\{0,1\}^3}|x\rangle$$

$$H^{\otimes 3}|\psi\rangle = \frac{1}{\sqrt{2^3}}\sum_x H^{\otimes 3}|x\rangle$$

$$= \frac{1}{\sqrt{2^3}}\sum_x \frac{1}{\sqrt{2^3}}\sum_y (-1)^{x\cdot y}|y\rangle$$

$$|x\rangle = |000\rangle$$
$$|001\rangle$$
$$\vdots$$

Asymptotic notations

$$O(\cdot), \Omega(\cdot), \Theta(\cdot), o(\cdot), \omega(\cdot)$$

Notation	Definition	Think	Example
$f(n) = O(g(n))$	$\exists c > 0, n_0 > 0, \forall n > n_0$: $0 \leq f(n) \leq cg(n)$	Upper bound	$100n^2 = O(n^3)$
$f(n) = \Omega(g(n))$	$\exists c > 0, n_0 > 0, \forall n > n_0$: $0 \leq cg(n) \leq f(n)$	Lower bound	$100n^2 = \Omega(n^{1.5})$
$f(n) = \Theta(g(n))$	$f(n) = O(g(n))$ & $f(n) = \Omega(g(n))$	Tight bound	$\log(n!) = \Theta(n \log n)$
$o(\cdot), \omega(\cdot)$		Strict upper/lower bound	$n^2 = o(2^n)$ $n^2 = \omega(\log n)$

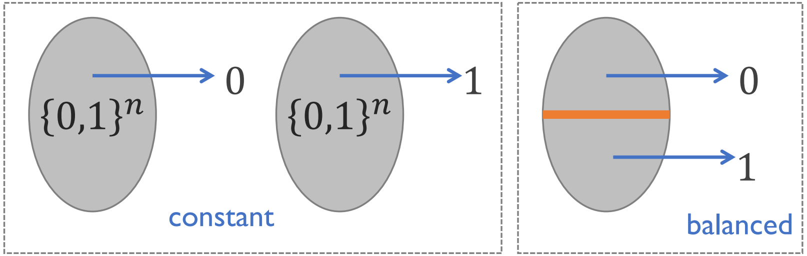
$\Omega(1)$ constant 1000, 3/4 ...

Reflection on Deutsch-Josza

Given: black-box $f: \{0,1\}^n \to \{0,1\}$ either constant or balanced

- constant means $f(x) = 0$ for all x, or $f(x) = 1$ for all x
- balanced means $\Sigma_x f(x) = 2^{n-1}$

Goal: decide which case



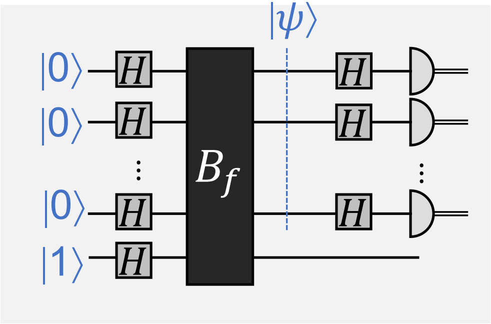
- Consider all $f: \{0,1\}^{\cancel{n}\,3} \to \{0,1\}$
  - # of constant functions $\underline{2}$
  - # of balanced functions $\underline{70}$ $\quad 2^3 = 8$
  - Total # of functions $\underline{256}$ $\quad (2^{2^n})$
- This is called a $\boxed{\text{Promise}}$ problem

$$\binom{8}{4} = \frac{8!}{4!\,(8-4)!} = \frac{8\times 7\times 6\times 5}{4\times 3\times 2\times 1} = 70$$

x	f(x)
000	
001	
⋮	
111	

$$\underbrace{2\cdot 2\cdot \cdots \cdot 2}_{8} = 2^8 = 256$$

Reflection on Deutsch-Josza



$$|\psi\rangle \propto \begin{cases} H^{\otimes n} \left(\pm \sum_{x\in\{0,1\}^n} |x\rangle \right), & \to |0^n\rangle \quad f \text{ constant} \\ \text{orthogonal to } \left(\pm \sum_x |x\rangle\right), & f \text{ balanced} \quad \to |x \neq 0^n\rangle \end{cases}$$

How to distinguish between the two cases?

What is $H^{\otimes n}|\psi\rangle$?

- Constant: $H^{\otimes n}|\psi\rangle = \pm|00\ldots0\rangle$
- Balanced: $H^{\otimes n}|\psi\rangle \in (\pm|00\ldots0\rangle)^{\perp}$

Simon's algorithm

Quantum vs. classical separations

Black-box problem	Classical deterministic	Randomized $\Omega(1)$ prob.	Quantum
Deutsch (1-bit constant vs. balanced)	2 (queries)	2 (queries)	1 (query)
Deutsch-Josza (n-bit constant vs. balanced)	$2^{n-1}+1$	$\Omega(n)$	1 Exact
Simon	$2^{n-1}+1$	$\Omega(\sqrt{2^n})$	$O(n)$ $\Omega(1)$ prob.

Simon's problem

Given: a black-box function $f: \{0,1\}^n \to \{0,1\}^n$

- **Promise:** there exists secret $s \neq 0^n$ such that
  $\forall x \neq x' \in \{0,1\}^n, f(x) = f(x')$ iff. $x \oplus x' = s$

Goal: find secret string s.

x	$f(x)$
$x_1, x_1 \oplus s$	
$x_2, x_2 \oplus s$	
…	
$x_k, x_k \oplus s$	
…	

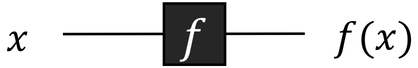
Example.

x	$f(x)$
000	011
001	101
010	000
011	010
100	101
101	011
110	010
111	000

$111 \oplus 010 = 101$

What is s in this case? ________

Classical algorithms for Simon

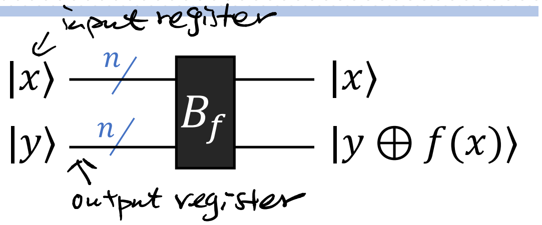


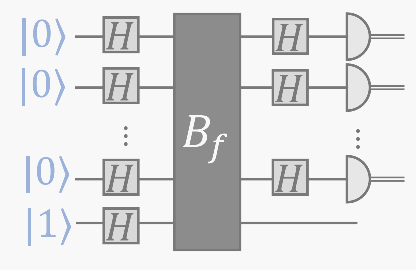
- Search for a collision: an $x \neq y$ such that $f(x) = f(y)$
  - Choose $x_1, x_2, \ldots, x_k \in \{0,1\}^n$ randomly (independently)
  - For all $i \neq j$, if $f(x_i) = f(x_j)$, then output $x_i \oplus x_j$ and halt
- A hard case: s is chosen at random & $f(x)$ is chosen randomly subject to the structure implied by s
- Birthday bound: $k = \Theta(\sqrt{2^n})$ to see a collision with constant (e.g., 3/4) probability
- This strategy is essentially optimal. (NB. You have to rule out all possible randomized algorithms)

A quantum algorithm for Simon

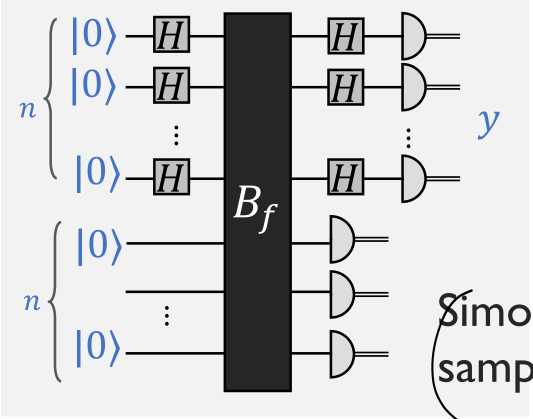
Recall: quantum black-box function

Unitary $B_f: |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$





Deutsch-Josza



Simon's quantum sampling subroutine

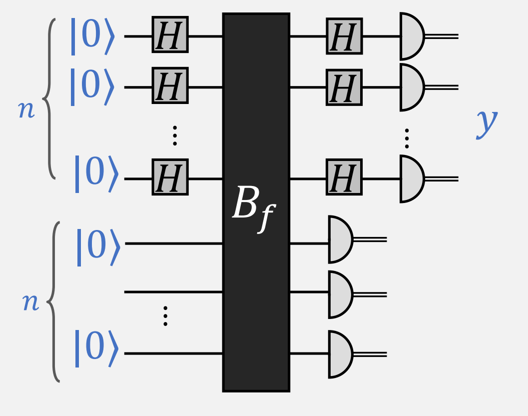
Simon's algorithm

1. Run Simon's quantum sampling subroutine k times.

Obtain samples $y_1, \ldots, y_k$

2. Post-processing. Classical

Solving linear system to find s



Theorem. $k = O(n)$ quantum queries suffice to find s w. prob. $\geq 1/4$.

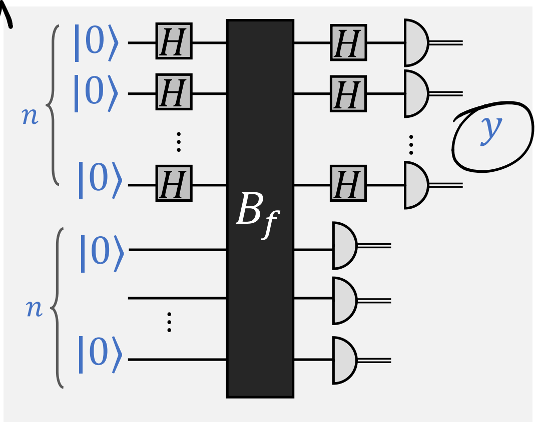
Simon's algorithm: analysis

1. Run Simon's quantum sampling subroutine k times.

   Obtain samples $y_1, \ldots, y_k \in \{0,1\}^n$

2. Classical post-processing on $\{y_i\}$.

   Solving linear system to find s



a. What do the samples y_i tell us?

b. How many samples are needed?

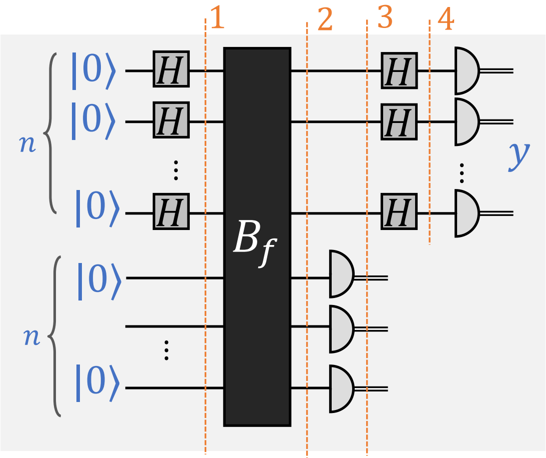
Remarks on notations

- $xy, \alpha\beta, AB$ usually denotes multiplication (integers, complex numbers, matrices)
- Strings $x, y \in \{0,1\}^n$, $x \cdot y$ denotes dot product, i.e., sum of bit-wise mult. mod 2 (for single bit: $x + y \bmod 2 = x \oplus y, x \cdot y = xy$)
- Concatenation $x||y$

Simon's algorithm: analysis I

a. What do the samples y_i tell us?

$n=3 \quad |x\rangle = |000\rangle$



$$|0^n\rangle|0^n\rangle$$

$$\xmapsto[1]{H^{\otimes n}\otimes \mathbb{1}} \left(\frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n} |x\rangle\right)|0^n\rangle$$

$$= \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|0^n\rangle$$

$$\xmapsto{B_f} \frac{1}{\sqrt{2^n}}\sum_x B_f(|x\rangle|0^n\rangle)$$

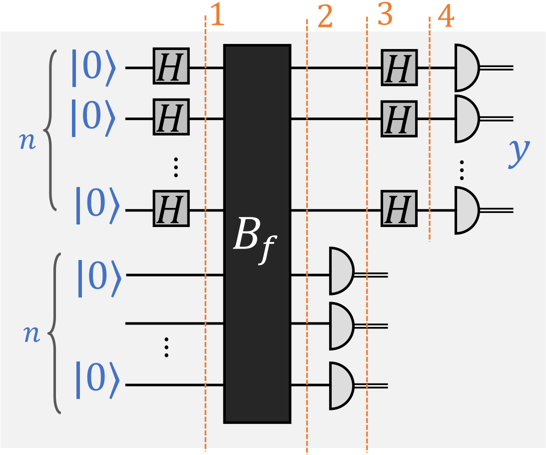
$$B_f : |x\rangle|y\rangle \longmapsto |x\rangle|y\oplus f(x)\rangle$$

$$\overset{2}{=} \frac{1}{\sqrt{2^n}}\sum_x |x\rangle|f(x)\rangle$$

Simon's algorithm: analysis I

a What do the samples y_i tell us?

$n=3 \quad |x\rangle = |000\rangle$



$\overset{2}{=} \frac{1}{\sqrt{2^n}} \sum_x |x\rangle \underline{|f(x)\rangle}$

3 meas bottom n qubits

observe	w.p.	posterior state
$a \in \{0,1\}^n$	$\cancel{1/2^{n-1}}$	$\frac{1}{\sqrt{2}}(\lvert x_a\rangle + \lvert x_a \oplus s\rangle)$

· which terms contribute to "a"?

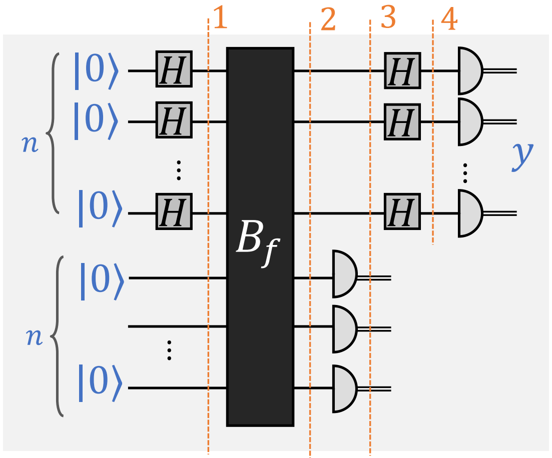
i.e. $f^{-1}(a) = \{x_a, x_a \oplus s\}$

$$\frac{\frac{1}{\sqrt{2^n}}(|x_a\rangle + |x_a \oplus s\rangle)|a\rangle}{\sqrt{\left(\frac{1}{2^n}+\frac{1}{2^n}\right)}}$$

$B_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$

Simon's algorithm: analysis I

a What do the samples y_i tell us?



3 meas bottom n qubits $\longmapsto$

Observe	w.p.	posterior state
$a \in \{0,1\}^n$	$1/2^{n-1}$	$\frac{1}{\sqrt{2}}(\lvert x_a\rangle + \lvert x_a \oplus s\rangle)$

$$\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2}}\left(H^{\otimes n}\lvert x_a\rangle + H^{\otimes}\lvert x_a \oplus s\rangle\right)$$

$$= \frac{1}{\sqrt{2^{n+1}}}\left(\sum_{y\in\{0,1\}^n} (-1)^{x_a \cdot y}\lvert y\rangle + \sum_{y\in\{0,1\}^n} (-1)^{(x_a\oplus s)\cdot y}\lvert y\rangle\right)$$

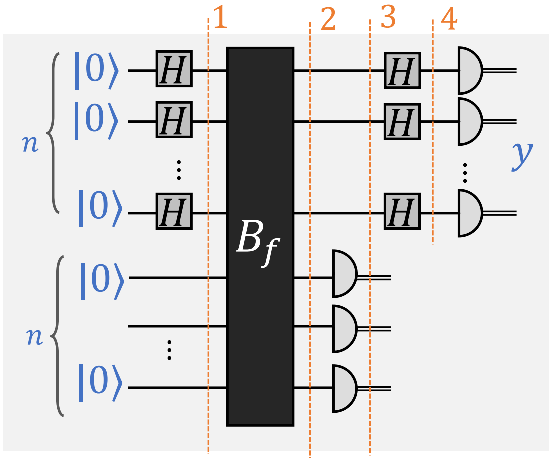
4 $$= \frac{1}{\sqrt{2^{n+1}}} \sum_{y\in\{0,1\}^n} \left((-1)^{x_a\cdot y} + (-1)^{(x_a\oplus s)\cdot y}\right)\lvert y\rangle$$

$$B_f : \lvert x\rangle\lvert y\rangle \longmapsto \lvert x\rangle\lvert y \oplus f(x)\rangle$$

$$H^{\otimes n}\lvert x\rangle = \frac{1}{\sqrt{2^n}}\sum_y (-1)^{x\cdot y}\lvert y\rangle$$

Simon's algorithm: analysis I

a What do the samples y_i tell us?



$$4 = \frac{1}{\sqrt{2^{n+1}}} \sum_{y \in \{0,1\}^n} \left((-1)^{x_a \cdot y} + (-1)^{(x_a \oplus s)\cdot y} \right) |y\rangle$$

$$= \sum_{y \in \{0,1\}^n} \alpha_y \, |y\rangle ,$$

$$\alpha_y := \frac{1}{\sqrt{2^{n+1}}} \left((-1)^{x_a \cdot y} + (-1)^{(x_a \oplus s)\cdot y} \right)$$

$\xmapsto{\text{meas.}}$

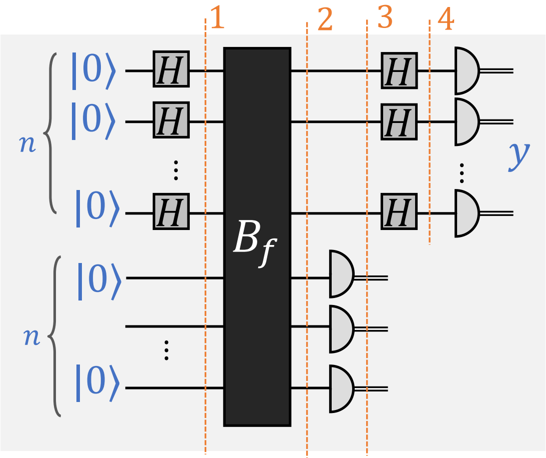
Obs.	w.p.	posterior state
"y"	$\lvert \alpha_y \rvert^2$	$\lvert y\rangle$

$$B_f : |x\rangle|y\rangle \longmapsto |x\rangle|y \oplus f(x)\rangle$$

$$H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_y (-1)^{x\cdot y} |y\rangle$$

Simon's algorithm: analysis I

a What do the samples y_i tell us?



$$(x_a \oplus s)\cdot y = x_a \cdot y \oplus s \cdot y$$

$$\alpha_y := \frac{1}{\sqrt{2^{n+1}}}\left((-1)^{x_a\cdot y} + (-1)^{(x_a\oplus s)\cdot y}\right)$$

$\xmapsto{\text{meas.}}$

obs.	w.p.	posterior state
"y"	$\lvert\alpha_y\rvert^2$	$\lvert y\rangle$

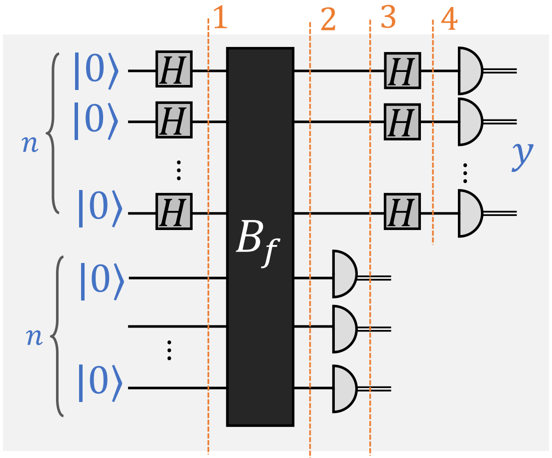
$$\Pr[y] = |\alpha_y|^2 = \frac{1}{2^{n+1}}\left|(-1)^{x_a\cdot y} + (-1)^{(x_a\oplus s)\cdot y}\right|^2$$

$$= \frac{1}{2^{n+1}}\left|(-1)^{x_a\cdot y} + (-1)^{x_a\cdot y}\cdot(-1)^{s\cdot y}\right|^2$$

$$= \frac{1}{2^{n+1}}\left|1 + (-1)^{s\cdot y}\right|^2$$

Simon's algorithm: analysis I

a What do the samples y_i tell us?



$(x_a \oplus s)\cdot y = x_a \cdot y \oplus s\cdot y$

$\alpha_y := \frac{1}{\sqrt{2^{n+1}}}\left((-1)^{x_a\cdot y} + (-1)^{(x_a \oplus s)\cdot y} \right)$

$\xmapsto{\text{meas.}}$

obs.	w.p.	posterior state
"y"	$\lvert\alpha_y\rvert^2$	$\lvert y\rangle$

$\Pr[y] = \lvert\alpha_y\rvert^2 = \frac{1}{2^{n+1}} \left\lvert 1 + (-1)^{s\cdot y} \right\rvert^2$

Case 1: $y\cdot s = 1$ $\quad \lvert\alpha_y\rvert^2 = 0$

Case 2: $y\cdot s = 0$ $\quad \lvert\alpha_y\rvert^2 = \frac{1}{2^{n-1}}$

what "y" we can see?

- $y\cdot s = 0$ a random y

Simon's algorithm: analysis II

b How many samples are needed?

$$\begin{aligned} y_1 \cdot s &= 0 \\ y_2 \cdot s &= 0 \\ &\cdots \\ y_k \cdot s &= 0 \end{aligned} \quad \Leftrightarrow \quad \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k1} & y_{k2} & \cdots & y_{kn} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

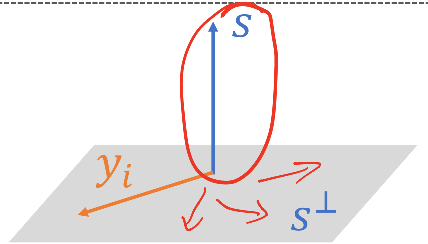
Fact. When $k = n - 1$, unique solution s with prob. $\geq \frac{1}{4}$

$$\Pr[y_1, \ldots, y_{n-1} \text{ linearly indep.}] \geq 1/4$$

Efficient algorithm: $O(n^{2.376})$ Coppersmith-Winograd

Simon's algorithm: a geometric interpretation

- Viewing $\{0,1\}^n$ as a vector space
  - $\mathbb{Z}_2 := \{0,1\}$ with addition and multiplication mod 2 is a field
  - $\{0,1\}^n = \mathbb{Z}_2 \times \cdots \times \mathbb{Z}_2 = \mathbb{Z}_2^n$ is an n-dimensional vector space over $\mathbb{Z}_2$
- Let $x \cdot y = x_1 y_1 + \cdots x_n y_n \bmod 2$ "dot product"
  - $\boxed{x \cdot y = 0}$ can be interpreted as the vectors being "orthogonal" (Not precise: e.g., $\exists x \neq 0, x \cdot x = 0$)



"orthogonal" complement

$s^\perp := \{y : y \cdot s = 0\}$

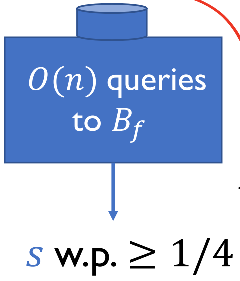
- Quantum sampling subroutine samples from $s^\perp$ uniformly at random
- $O(n)$ independent samples determines s with constant probability

Recap: quantum speedups

Black-box problem	Classical deterministic	Randomized $\Omega(1)$ prob.	Quantum
Deutsch (1-bit constant vs. balanced)	2 (queries)	2 (queries)	1 (query)
Deutsch-Josza (n-bit constant vs. balanced)	$2^{n-1}+1$	$\Omega(n)$	1 Exact
Simon	$2^{n-1}+1$	$\Omega(\sqrt{2^n})$	$O(n)$ $\Omega(1)$ prob.

exponential speed up

Exercise: amplifying the success probability



- How to find s with probability $\geq 1 - 2^{-n}$?
- How many quantum queries will be needed?

biased $H: 0 < \varepsilon < \frac{1}{2}$

$T: 1 - \varepsilon$

$\Pr[\text{BAD}]$

$= \Pr[\text{COIN 1 TAILS}] \cdot \Pr[\text{2 TAILS}] \cdots \Pr[\text{COIN } m \text{ Tails}]$

1 2 … m

indep. $= (1-\varepsilon)^m$

Goal: see at least one HEADS

→ BAD: all coins in T

$\Pr[\text{Succ}] = 1 - (1-\varepsilon)^m$

$\varepsilon = 1/4 \quad m = n$

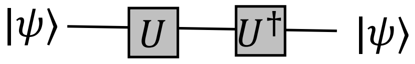
Reversible computation

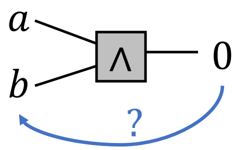
Quantum vs. classical computation

- We've seen a few examples where quantum algorithms outperform classical ones → quantum computer is powerful
- But, wait a second, we haven't even justified a basic goal …

> Is a quantum computer (at least) as powerful as a classical computer?
> i.e. can an arbitrary efficient classical algorithm (circuit) be converted to an efficient quantum algorithm (circuit)?

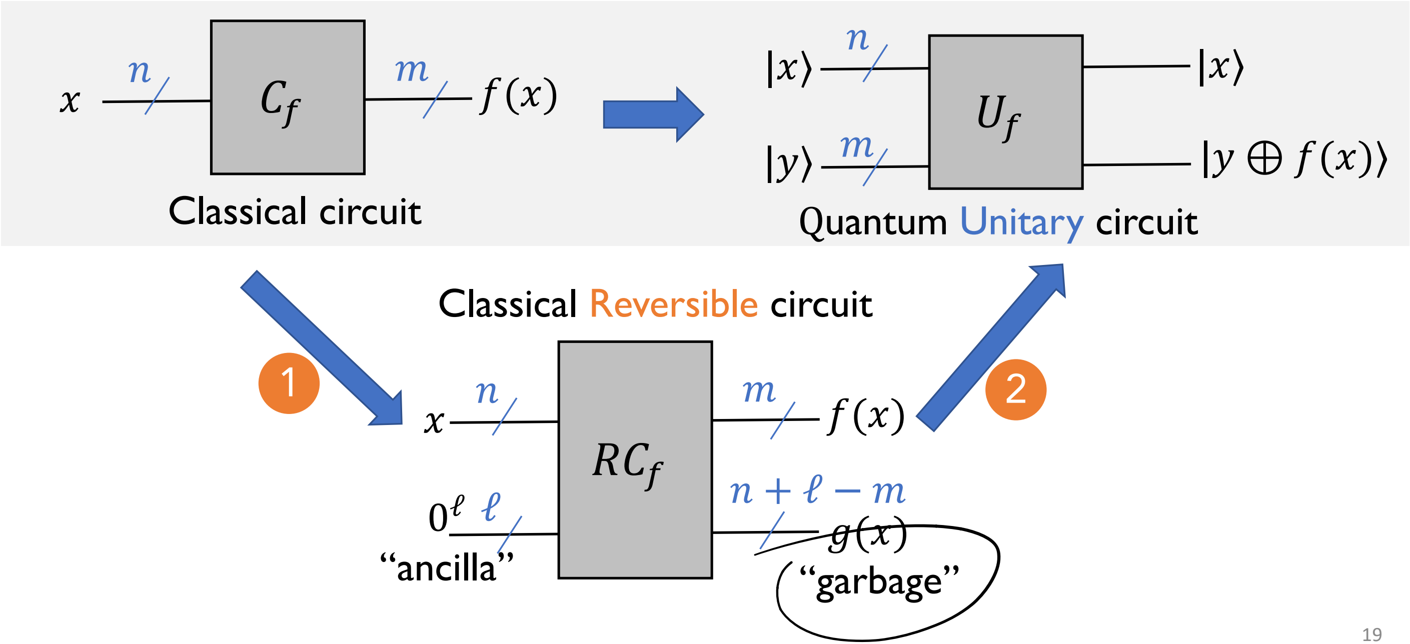
- Not immediate, quantum ckt (w.o. meas.) is unitary ➔ reversible



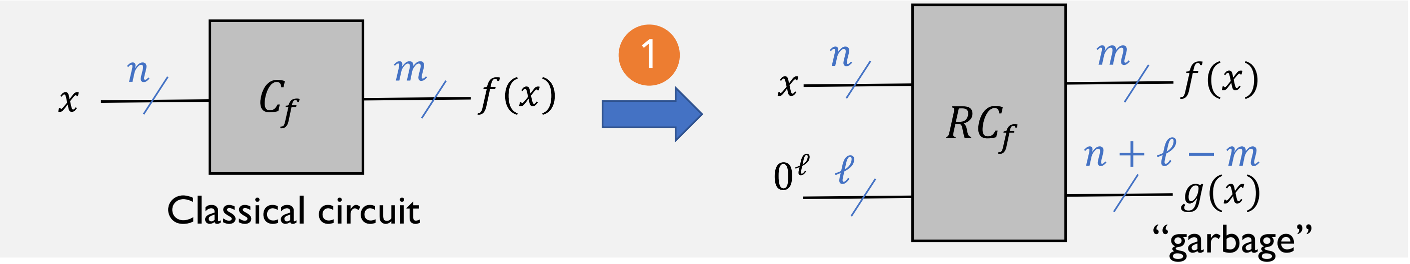


Simulating classical circuit

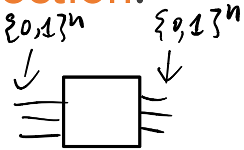
- Consider $f: \{0,1\}^n \to \{0,1\}^m$ Ex. $f(x_1, x_2) = x_1 + x_2 \bmod 2^m, n = 2m$

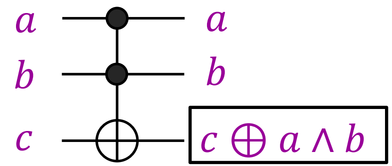


1. Making classical circuit reversible



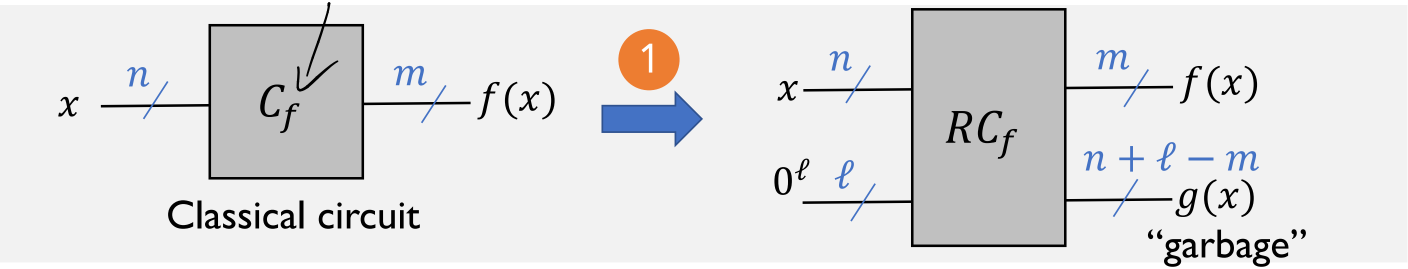
- Def. A Boolean gate is reversible if it has the same input / output size, and the input to output mapping is a bijection.



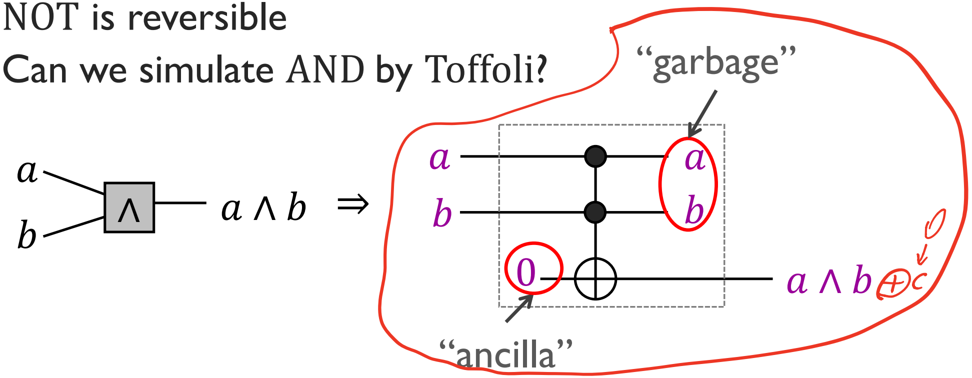


Toffoli gate
(controlled-controlled NOT)
"flip c iff. Both a and b are 1"

1. Making classical circuit reversible

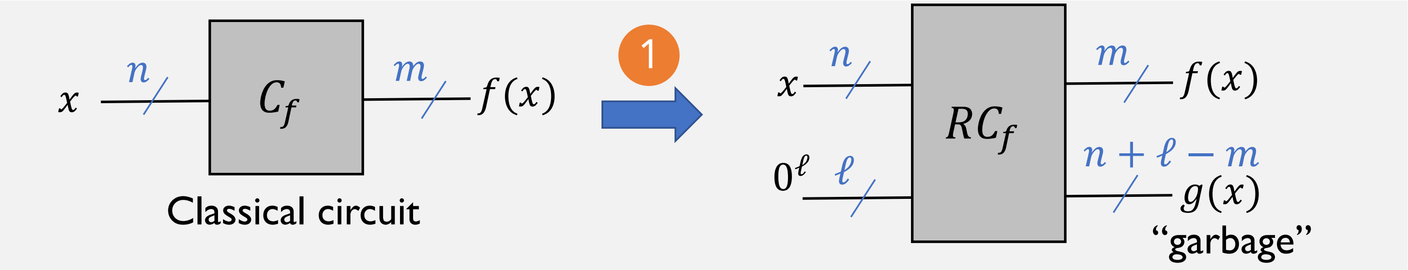


- **Fact.** {AND, NOT} gates are universal for classical circuits
  - NOT is reversible
  - Can we simulate AND by Toffoli?



NAND

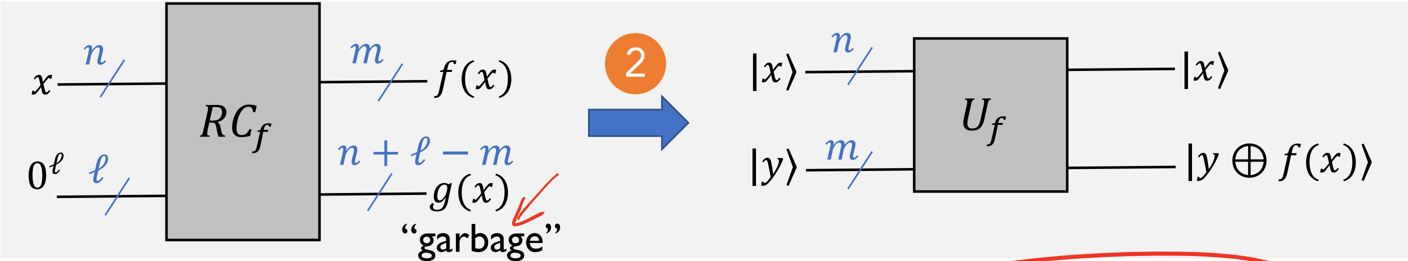
1. Making classical circuit reversible



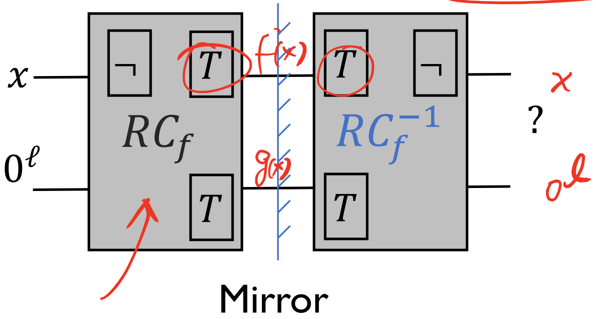
Replace each AND with reversible Toffoli gadget

$$|C_f| = k \longrightarrow |RC_f| = O(k)$$

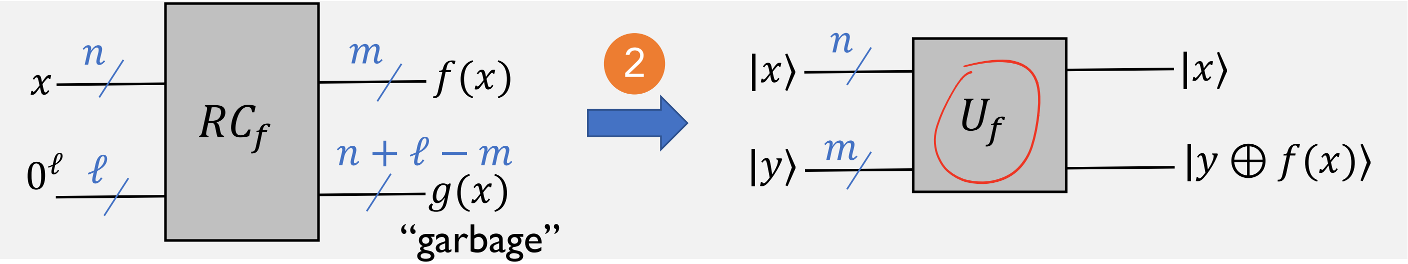
2. Cleaning up the junk



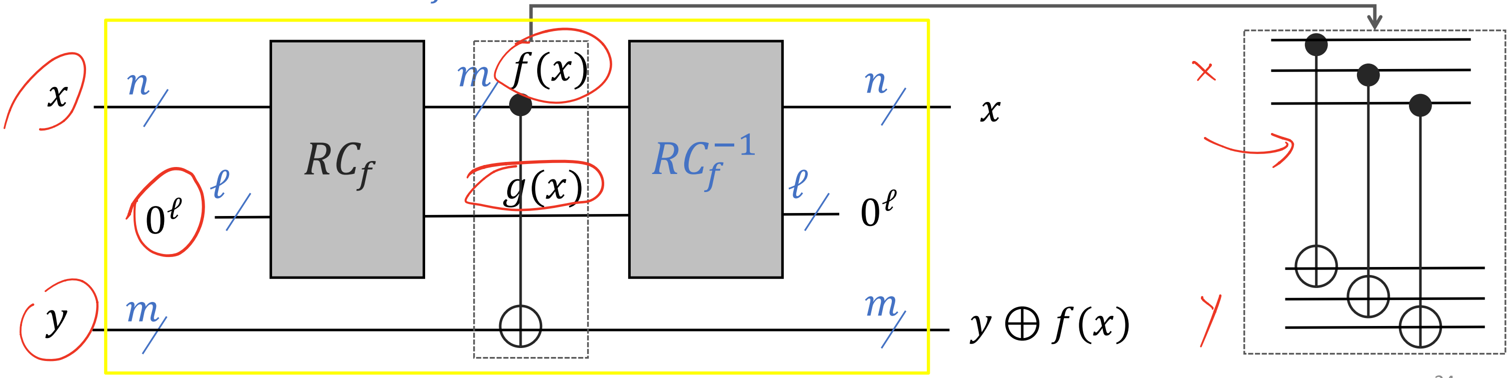
- What does the "mirror" of RC_f do? – it uncomputes



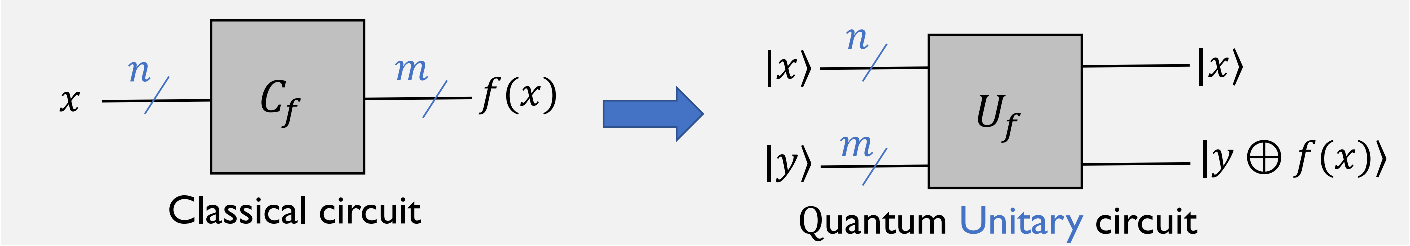
2. Cleaning up the junk



Quantum circuit U_f: $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$ $\quad |U_f| = 2|RC_f| + m$



Summary



$$|C_f| = k \longrightarrow |RC_f| = O(k) \longrightarrow |U_f| = O(k)$$

Corollary. $\mathbf{BPP} \subseteq \mathbf{BQP}$ [More to come in future]
Any problem that a classical computer can solve efficiently can be solved on a quantum computer efficiently too

Logistics

- HW3 due Sunday

- Project
  - Project page: instructions and suggested topics
  - Send me your group information by end of today (April 24 11:59pm AoE).
  - Proposal due next Sunday May 3rd , 11:59pm AoE.
  - Ask for feedback and start brainstorming (e.g., Campuswire private chat rooms)
  - End of today's lecture: group discussion

Project discussion

CCC report

- Quantum algorithms
  - List 3 major algorithm design directions
  - What is the prospect of the timeline for quantum algorithms?
- Quantum computing architecture
  - List three major considerations facing a quantum architecture design
- Quantum programming
  - What is the focus of current effort and what future effort would be needed?
- Verification
  - What are the different levels of verification? What tools are needed?

Scratch