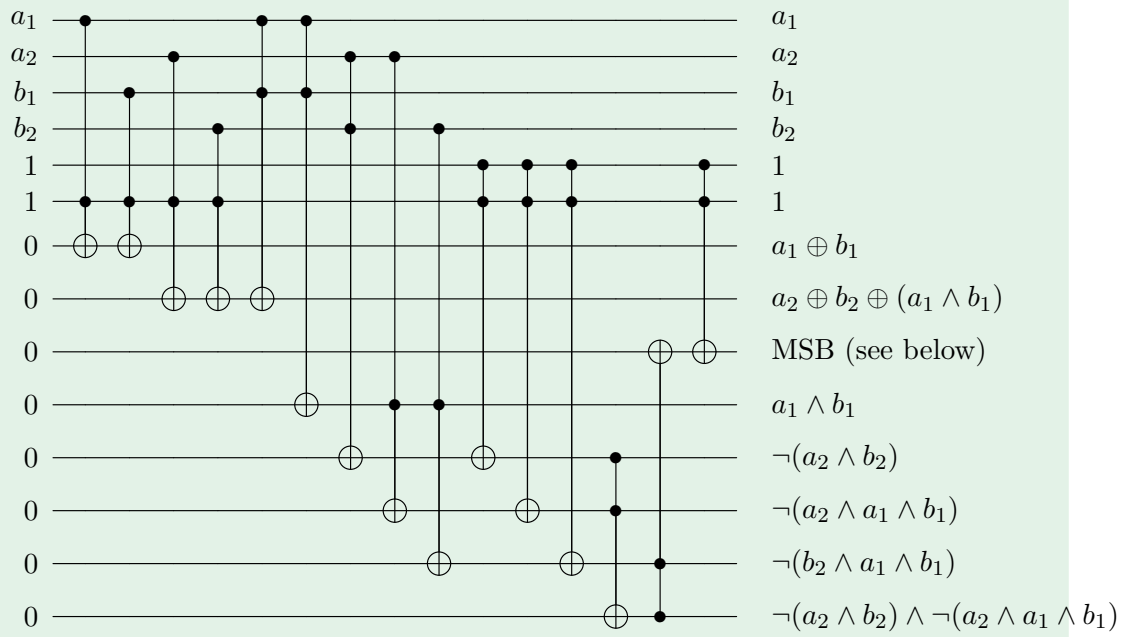
Introduction to Quantum Information Science
Homework 9

Due Wednesday, November 10 at 11:59 PM

Note: You should explain your reasoning, i.e. show your work, for all problems. You do not need to show us every step of each calculation, but every answer should include an explanation *written with words* of what you did.

1. Toffoli-based Addition [6 Points] Work out an explicit circuit of Toffoli gates for adding two 2-bit integers to get a 3-bit integer — assume the integers are unsigned, encoded in binary in the usual, simplest way. You can use arbitrary ancilla bits initialized to 0 or 1. Be sure to designate your input, output, and garbage registers.
Show the garbage bits that are generated by your circuit when 11 is added to 10.

Solution: The following circuit adds two integers a_2a_1 and b_2b_1. The first 4 bits are inputs, the next 2 bits are ancilla bits initialized to 1, the next 3 bits are the output (in order of least significant bit to most significant bit), and the remaining bits are garbage.



Why does this work? The first two gates compute the least significant output bit, which is $a_1 \oplus b_1$. The next three gates compute the second least significant output bit, which is $a_2 \oplus b_2 \oplus (a_1 \wedge b_1)$ (notice that $a_1 \wedge b_1$ is essentially the carry bit from the adding the least significant bits). The most significant bit (MSB) is the carry bit of the addition of the previous two bits, which is 1 if either $a_2 \wedge b_2$ or $a_2 \wedge (a_1 \wedge b_1)$ or $b_2 \wedge (a_1 \wedge b_1)$ (i.e. if at least two of the three bits being added are 1). The

remaining Toffoli gates compute the OR of these three expressions using DeMorgan's Law.

When $a_2a_1 = 11$ and $b_2b_1 = 10$, we can see that the garbage bits are, respectively: 00110.

2. The Birthday Paradox Your favorite local radio station is running a new give-away contest that works as follows: Each day at 5pm the phone lines open up to the public to call in and leave their name and birth-date which is then added to a running list. The contest runs until a matching birth-date (month and date) between any two contestants on the list is found. At that point the contest closes and everyone on the list up to that point is a winner. Assume that every birthday is equally likely and that no contestants are born during a leap year (so that we can ignore Feb. 29^{th}).

As usual, you must show or explain your work for each part. You are free to use numerical software of your choice to help solve the problem

a) [2 Points] What is the minimum number of people who need to call in before the probability of a match being found is at least 50%?

Solution: One possible approach to solving this problem is to make use of a little mathematical induction. We know that the probability that any two people share the same birthday is $p = \frac{1}{365}$ and likewise the probability that two people don't share the same birthday is $\bar{p} = 1 - p = \frac{364}{365}$.

General Sol'n: Say we have n people, what is the probability that none of them share the same birthday?

$n = 2 \qquad \bar{p} = \dfrac{364}{365}$

$n = 3 \qquad \bar{p} = \dfrac{364}{365} \times \dfrac{363}{365}$ Probability that the first two people don't share the same birthday times the probability that the third person doesn't share a birthday with the first two.

$n = 4 \qquad \bar{p} = \dfrac{364}{365} \times \dfrac{363}{365} \times \dfrac{362}{365}$ Probability that the first three people don't share the same birthday times the probability that the fourth person doesn't share a birthday with the first three.

$\vdots \qquad\qquad \vdots$

$$n \quad \rightarrow \quad \bar{p}(n) = \frac{365 \times 364 \times 363 \times \cdots \times (365 - n + 1)}{365^n} = \frac{365!}{365^n(365 - n)!}$$

Now we can use the above formula to solve for the value of n for which $\bar{p} = .5$. Unfortunately the above result is a transcendental equation– try applying Stirling's approximation to the factorials and working it out if you want to see it– so a direct analytical solution is almost certainly not possible. There's a few possible ways forward (Taylor expansions etc...) but we promised we wouldn't use calculus so we'll do it numerically using the built in root finder in Mathematica (FindRoot).

$$.5 = \frac{365!}{365^n(365 - n)!} \quad \rightarrow \quad n \approx 22.77$$

So rounding up to the next person the contest has a little more than 50% chance of ending when at least 23 contestants have called in.

b) [1 Point] How about the minimum number of people needed before there is a 99% chance of the contest coming to a close?

Solution: We can use the same formula that we derived in part a but now set $\bar{p} = .01$ and again use the numerical solver.

$$.01 = \frac{365!}{365^n(365-n)!} \quad \rightarrow \quad n \approx 56.92$$

So after 57 people have called in there is a 99% chance that the contest will have come to a close.

c) [4 Points] Imagine after running the contest for a few days the station decides they aren't being generous enough and so change the rules as follows: Instead of the contest ending as soon as there's a match found between any two contestants on the list it now ends when a match is found between specifically the first caller of the day and any other contestant.

Under these new rules what is the minimum number of people needed before there is a 50% chance of the contest closing? How about for a 99% chance?

Solution: The probability that any given contestant shares a birthday with the first caller is as before $p = \frac{1}{365}$. With a list of n people the probability that there will not be a match specifically between the first caller and the remaining is:

$$\bar{p}(n) = \left(\frac{364}{365}\right)^{n-1}$$

Note: We need to be careful not to accidentally include matchings between the first caller and his own birthday, hence the exponent of $n-1$ rather than n.

50% chance of matching:

$$\bar{p}(n) = .5 = \left(\frac{364}{365}\right)^{n-1}$$

$$\ln(.5) = (n-1)\ln\left(\frac{364}{365}\right)$$

$$n = \frac{\ln(.5)}{\ln(\frac{364}{365})} + 1$$

$$n \approx 253.65$$

After 254 people have called in there is a 50% chance of the contest coming to a close.

99% chance of matching:

$$\bar{p}(n) = .01 = \left(\frac{364}{365}\right)^{n-1}$$

$$\ln(.01) = (n-1)\ln\left(\frac{364}{365}\right)$$

$$n = \frac{\ln(.01)}{\ln(\frac{364}{365})} + 1$$

$$n \approx 1679.58$$

After 1680 people have called in there is a 99% chance of the contest coming to a close.

3. Two Secrets [5 Points] Suppose the function $f : \{0,1\}^n \to \{0,1\}^n$ is such that there are *two* n-bit secret strings s and t, where that $s \neq t$ and neither are the all zeros string, so that $f(x) = f(y)$ if and only if $x \oplus y$ is either 0, s, t, or $s \oplus t$.
Give a quantum algorithm which can find an $a \in \{0,1\}^n$ such that $a \cdot s = a \cdot t = 0$. Explain/prove that it works.

Solution: Another way of writing the constraint is

$$f(x) = f(x \oplus s) = f(x \oplus t) = f(x \oplus s \oplus t).$$

This means that if we run the same Simon's problem algorithm, after we do the first partial measurement on the ancilla registers (after the first set of hadamards and calling the oracle), we're left with the state

$$\frac{1}{2}(|x\rangle + |y\rangle + |z\rangle + |w\rangle)$$

such that WLOG $x \oplus y = s$, $x \oplus z = t$, and $x \oplus w = s \oplus t$. Measuring this state in the Hadamard basis gives:

$$H^{\otimes n}\frac{1}{2}(|x\rangle + |y\rangle + |z\rangle + |w\rangle) = \frac{1}{2^{n/2+1}} \sum_{a \in \{0,1\}^n} \left[(-1)^{x\cdot a} + (-1)^{y\cdot a} + (-1)^{z\cdot a} + (-1)^{w\cdot a}\right] |a\rangle .$$

For the amplitude on $|a\rangle$ to be non-zero, we need

$$x \cdot a + y \cdot a + z \cdot a + w \cdot a \neq 2.$$

This means that either 3 or all 4 terms are equal to one another. If all 4 are equal then

$$x \cdot a = (x \oplus s) \cdot a = (x \oplus t) \cdot a = (x \oplus s \oplus t) \cdot a \Rightarrow 0 = s \cdot a = t \cdot a = (s \oplus t) \cdot a$$

If 3 are equal then WLOG (due to symmetry)

$$x \cdot a = (x \oplus s) \cdot a = (x \oplus t) \cdot a \Rightarrow 0 = s \cdot a = t \cdot a$$

4. The Quantum Fourier Transform The Quantum Fourier Transform QFT_d is a quantum gate acting on *qudits*, i.e. quantum systems with d levels. It is defined below for $x, y \in \{0, 1, ..., d-1\}$.

$$QFT_d |x\rangle = \frac{1}{\sqrt{d}} \sum_{y=0}^{d-1} \omega^{xy} |y\rangle$$

where $\omega = e^{2\pi i/d}$ is a primitive dth root of unity.

To be clear, a qudit is just like a qubit but it's a vector of d amplitudes instead of just 2 amplitudes. Then, a unitary acting on a qudit has dimension $d \times d$.

a) [3 Points] Calculate QFT_2, QFT_3, and QFT_4 explicitly (either by writing down the corresponding matrix or equivalently by specifying the action on each of the standard basis states). By what other name is QFT_2 known?

Solution:

$$QFT_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \to \quad \text{This is the Hadamard gate.}$$

$$QFT_3 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{i(2/3)\pi} & e^{i(4/3)\pi} \\ 1 & e^{i(4/3)\pi} & e^{i(2/3)\pi} \end{bmatrix}; \quad QFT_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

b) [3 Points] Prove that QFT_d is unitary for all d.

Solution:
Observe that $QFT_d = QFT_d^T$ because each matrix element depends only on $xy = yx$. Calculate matrix elements of $QFT_d^\dagger QFT_d$:

$$\langle z| QFT_d^\dagger QFT_d |x\rangle = \langle z| QFT_d^\dagger \frac{1}{\sqrt{d}} \sum_{y=0}^{d-1} e^{2\pi i xy/d} |y\rangle$$

$$= \frac{1}{d} \sum_{w=0}^{d-1} \sum_{y=0}^{d-1} e^{2\pi i xy/d} e^{-2\pi i wy/d} \langle z|w\rangle = \frac{1}{d} \sum_{y=0}^{d-1} e^{2\pi i (x-z)y/d}$$

If $x = z$ then $\frac{1}{d}\sum_{y=0}^{d-1} e^{2\pi i(x-z)y/d} = \frac{1}{d}\sum_{y=0}^{d-1} 1 = d/d = 1$. If $x \neq z$ then the sum cycles over all d'th roots of unity which sum to 0. Thus $QFT_d^\dagger QFT_d = I$.

c) [3 Points] For which values of d is QFT_d its own inverse?

Solution: Following the above, the matrix elements are now:

$$\langle z| QFT_d QFT_d |x\rangle = \frac{1}{d} \sum_{y=0}^{d-1} e^{2\pi i (x+z)y/d}$$

If $x = z$ then we want $\frac{1}{d}\sum_{y=0}^{d-1} e^{2\pi i(x+z)y/d} = 1$. This is only true if $x + z$ is a multiple of d for all x, z. We see $x + z = 2z$ is a multiple 2, so this only works for $d = 2$ (and trivially for $d = 1$).

d) [Extra Credit, 3 Points] In Recitation 4, we saw that the qudit clock and shift matrices are respectively defined as $X_d |x\rangle = |x + 1 \bmod d\rangle$ and $Z_d |x\rangle = \omega^x |x\rangle$ for $x \in 0, 1, \ldots, d-1$. Show that

$$QFT_d^\dagger X_d QFT_d = Z_d^\dagger.$$

Solution: Because all of these matrices are invertible, we can equivalently show that $X_d QFT_d = QFT_d Z_d^\dagger$. We do so by showing that $X_d QFT_d |x\rangle = QFT_d Z_d^\dagger |x\rangle$ for all $x \in 0, 1, \ldots, d-1$.

$$X_d QFT_d |x\rangle = X_d \frac{1}{\sqrt{d}} \sum_{y=0}^{d-1} \omega^{xy} |y\rangle = \frac{1}{\sqrt{d}} \sum_{y=0}^{d-1} \omega^{xy} |y + 1 \bmod d\rangle$$

$$QFT_d Z_d^\dagger |x\rangle = QFT_d \omega^{-x} |x\rangle = \frac{1}{\sqrt{d}} \sum_{y'=0}^{d-1} \omega^{x(y'-1)} |y'\rangle$$

If we make the change of variables $y' = y + 1 \bmod d$ then we can see that these are the same.

5. RSA Two spies of an enemy nation are known to use the RSA crypto-system with public keys

$$N = 4668619 \quad \text{and} \quad e = 3.$$

And intelligence has revealed that

$$\varphi(N) = 4664296.$$

Now we've intercepted a message sent between the two spies showing that they intend to meet at a certain time:

$$m = 1202997$$

Let's arrange for them (our enemies) to miss each other by an hour.

[Note that decrypted messages are encoded in ASCII with two ***decimal*** *digits per character, as described on* `www.asciitable.com`*. You're of course free to use computer help.* `www.dcode.fr/modular-exponentiation` *and* `www.dcode.fr/ascii-code` *may prove useful. But please don't forget to show intermediate steps in the solutions.]*

a) [2 Points] Confirm that our intelligence is correct by using N and $\varphi(N)$ to determine the two factors p and q of N and checking that indeed $pq = N$. (Note that part (a) is not required in the parts that follow.)

Solution: We know that

$$\varphi(N) = (p-1)(q-1) = \underbrace{pq}_{=N} - p - q + 1 = 4664296.$$

Thus,

$$\varphi(N) - N - 1 = -p - q = -4324.$$

This gives us the two equations

$$pq = 4668619$$
$$p + q = 4324.$$

And from these we can derive the quadratic equation

$$p^2 - 4324p + 4668619 = 0.$$

Hence, by the quadratic formula

$$p = \frac{4324 \pm \sqrt{4324^2 - 4 \cdot 4668619}}{2} = 2237 \text{ or } 2087.$$

By symmetry, if we select one of these to be p, the other will be q. And indeed

$$2237 \cdot 2087 = 4668619 = N.$$

b) [1 Point] Meanwhile an analyst has gone ahead and done the work of determining

$$d = 3109531$$

to be the inverse of e in the multiplicative group $\mathbb{Z}^{\times}_{\varphi(N)}$. Verify that this is correct.

Solution: A simple calculation reveals that indeed

$$(e \cdot d) \bmod \varphi(N) = (3 \cdot 3109531) \bmod 4664296 = 1.$$

c) [1 Point] Decrypt the given message.

Solution: Raising m to the d'th power mod N produces

$$m^d \bmod N = 1202997^{3109531} \bmod 4668619 = 548077.$$

Decoding this reveals

$$54 \mapsto 6, \quad 80 \mapsto \text{P}, \quad 77 \mapsto \text{M}.$$

d) [1 Point] Encrypt a new message in the same format instructing to meet an hour later; we will send this to the second spy instead of the original message.

Solution: The new message should be "7PM" whose ASCII encoding is 558077. Encrypting this gives us

$$558077^3 \bmod 4668619 = 3706174.$$