Physics 4481-7681; CS 4812 Problem Set 5

Five problems (four pages), covers lectures 15–18, due Fri evening 12 Nov 2021 at 23:00

[Note: the problem sets in this course are not intended to be group projects. It is permitted to discuss with other students at intermediate stages, but anything turned in must be entirely your own. In addition, to be in compliance with the Cornell Academic Integrity Code (link to http://cuinfo.cornell.edu/aic.cfm at the bottom of course website from the outset), you should explicitly credit any discussions you’ve had.]

Problem 1: see Jupyter notebook

Problem 2: Discrete Fourier transform

In class (lecture 18, and notes linked from course website), we emulated the steps to factor the number \( N = 15 \) via period finding. Now turn to factoring \( N = 21 \):

a) First consider the function \( f(x) = b^x \mod 21 \), with \( b = 16 \). For experience with discrete Fourier transform, imagine starting from a state \( |\Phi\rangle = \frac{1}{\sqrt{32}} \sum_{x=0}^{31} |x\rangle |f(x)\rangle \). Note the number of inputs bits is \( n = n_o = 5 \), i.e., we have not yet doubled the number of input bits.

(i) Suppose we measure the “output” \( |f(x)\rangle \) as 4. In what state \( |\Psi\rangle \) will that leave the “input”? Writing the state in the form \( |\Psi\rangle = \sum_{x=0}^{31} \gamma(x)|x\rangle \), graph the (discrete) function \( \gamma(x) \) for values of \( x \) from 0 to 31.

(ii) Now calculate the discrete Fourier transform \( \tilde{\gamma}(y) = \frac{1}{\sqrt{32}} \sum_{x=0}^{31} e^{2\pi i xy/32} \gamma(x) \), and graph the values of the modulus-squared \( |\tilde{\gamma}(y)|^2 \) for continuous \( y \) from 0 to 32, and on the same plot mark the values of \( |\tilde{\gamma}(y)|^2 \) for \( y \) an integer from 0 to 31.

(iii) Imagine now measuring the state \( U_{FT}|\Psi\rangle = \sum_{y=0}^{31} \tilde{\gamma}(y)|y\rangle \). With what numerical values of the probability would the various non-zero values of \( y \) be measured?

(iv) The prescription to double the number of input bits is to ensure a significant probability of measuring \( |y/2^n - j/r| < 1/(2N^2) \) (lec17_wb#7). For the current \( n = n_o = 5 \), \( N = 21 \) and integer \( j \), what is the total probability of measuring \( y \) satisfying that inequality? Given that measuring \( y = 0 \) gives no information about \( r \), what is the probability, with \( y = 0 \) excluded, for measuring a \( y \) satisfying the inequality?

b) Now consider \( f(x) = b^x \mod 21 \), still with \( b = 16 \), but now with \( n = 2n_o = 10 \). Start from the state \( |\Phi\rangle = U_f H^{\otimes 10}|0\rangle_{10}|0\rangle_5 = \frac{1}{2^7} \sum_{x=0}^{2^{10}-1} |x\rangle |f(x)\rangle \).

(i) If the output qubits are again measured to be 4, in what state \( |\Psi\rangle \) does that leave the input qubits?

(ii) Now imagine applying \( U_{FT}|\Psi\rangle \) as in a.iii) above, and then measuring the input qubits. What is the probability of measuring a non-zero value of \( y \) satisfying \( |y/2^n - j/r| < 1/(2N^2) \) (for \( n = 10 \), \( N = 21 \) and any integer \( j \)).

(iii) Unlike a classical discrete Fourier transform, in the quantum case the measurement in part (ii) only reveals a single Fourier transform component. Presuming you didn’t already know the value of the period \( r \) of the above \( f(x) \), how could you infer the period from each of those possible measured values of \( y \)?
Problem 3: Period Finding and continued fractions

This illustrates the mathematics of the final (post-quantum-computational) stage of Shor’s period finding procedure, as described in lec 17. Make use of the theorem, cited in the text, that if \( j \) and \( r \) have no common factors, and \( x \) is an estimate for the fraction \( j/r \) that differs from it by less than \( 1/2r^2 \), then \( j/r \) will appear as one of the partial sums in the continued fraction expansion of \( x \).

a) Suppose you know that the integer \( r \) is less than \( 2^{10} - 1 = 1023 \) and \( y = 206820 \) is assumed within \( \frac{1}{2} \) of an integer multiple of \( 2^{20}/r \). What are the possible values of \( r \)?

b) i) Suppose you know that the integer \( r \) is less than 1023 and you’ve measured \( y = 330632 \), assumed to be within \( \frac{1}{2} \) of an integral multiple of \( 2^{20}/r \). What are the possible values of \( r \)? ii) Suppose you decided to measure again, finding \( y = 279620 \), again assumed to be within \( \frac{1}{2} \) of an integral multiple of \( 2^{20}/r \). Combining with the information from part i), what is the value of \( r \)?

[These numbers are small enough to find with an exhaustive computer search, but try to answer them by using continued fractions, with the equivalent of a (nonprogrammable) calculator. Since \( 2^{20} > 1023^2 \), in either question the fraction \( j/r \) for some integer \( j \) will necessarily be one of the partial sums of the continued fraction expansion of \( y/2^{20} \), where \( y \) is one of the cited integers. The partial sum with the largest denominator less than 1023 is the one of interest. Check your answer by multiplying the fraction \( j/r \) you find by \( 2^{20} \). Partial fraction expansions and partial sums (covered in lecture 16) and the successive \( a_0, a_1, a_2, \ldots \) are determined by repeatedly taking the integral part, subtracting it, and inverting what remains. Include both the numbers \( a_0, a_1, a_2, \ldots \) from the continued fraction expansions you generate, and the partial sums you examined to extract the values of \( r \) in a) and b).]

c) [Bonus, to be added] Values to be posted as .npy file

d) [Bonus, to be added] For this problem, use the values of \( N \) (2048 bits) and \( r \) given in ps4#5. additional values to be posted in above .npy file.
Problem 4:

The problems with exaggerated claims for “precompiled” factor circuits were mentioned in lecture 17 (as well as the reductio ad absurdum provided by Smolin et al. [arXiv:1301.7007] — “factoring” a 20,000 bit number using only a single qubit).

For any \( N = pq \), it is not difficult to find numbers \( b \) with small order mod\( N \), e.g., \( b^x \mod N = 1 \) for \( x = 2 \) or 4.

Determine what happens to the circuit diagram given in lec17_wb slide #3 if a) \( U^4 = 1 \) or b) \( U^2 = 1 \). Draw an equivalent circuit with a minimal number of qubits for each of those two cases.

c) For example, for the (129 decimal digit) RSA-129* \( N = pq \), confirm that

\[
b_0 = 3332450229620833608567286587437353883812759325772615602723936
\]

satisfies \( b_0^{32} \mod N = 1 \), and therefore that \( b_1 = b_0^8 \mod N \) satisfies \( b_1^4 \mod N = 1 \) (with \( b_1^2 \mod N \neq 1 \)), and \( b_2 = b_0^{16} \mod N \) satisfies \( b_2^2 \mod N = 1 \).

Confirm (using Euclidean algorithm for gcd) that \( b_1 \) or \( b_2 \) can be used to factor RSA-129. (Therefore a simple “precompiled” circuit calculating \( f(x) = b^x \mod N \) for \( b \) equal to \( b_0 \) or \( b_1 \) could be used to “factor” RSA-129 with current cloud computers.)

Problem 5:

In lec17 (slides 2,3), it’s shown how to efficiently calculate the periodic function \( f(x) = b^x \mod N \), highlighting that it can be done with a single execution of the procedure that does the multiple squarings (without need of a classical look-up table).

(i) Recall that the time for the period finding is dominated by the calculation of the modular exponent function itself (as also discussed in lecture 17). Considering the number of modular multiplications, and how many additions each involves, and how many operations are involved in each addition, explain why calculating the modular exponent is \( O(n^3) \) (where \( n \) is the number of digits).

ii) Given the above estimate of how the number of gates scales in the number of digits of the number to be factored, roughly how many total gates (modular exponentiation plus quantum Fourier transform) might be necessary to factor the 2048 bit RSA number (the largest).

Note: the details will depend on the implementation, this is just to estimate some rough number.

(iii) Use the argument given in the right hand column of p.2 of arXiv:1208.0928 (and using the 100ns measurement time) to estimate how long it would take to factor the 829 and 2048 bit (250 and 617 decimal digit) numbers in ps4#4A, if the quantum method were

---

* https://en.wikipedia.org/wiki/RSA_numbers#RSA-129
p = 349052951084765094914784961990389813341776146384938838490820577
q = 32769132993266709549961988190834461413177642967992942539798288533

3
currently feasible.
(The point here is just to confirm that $n^3$ grows much more slowly than $\exp(\alpha n^{1/3})$.
The method in the article results in an estimate of $40n^3$ for the scaling of the number of operations to calculate the modular exponent function.)

(iv) arXiv:1905.09749 was also mentioned in class (Lec 12), and finds the period of a slightly different function than Shor’s $b^x \mod N$ (see sec 2.A. of the article, using classical number theory). This method still permits factoring $N$, but with fewer overall multiplications, reducing the coefficient of the $n^3$ term in part (iii) to $.3n^3$ (see Table I in the article). What is the effect on the results of part (iii) for the two values of $N$?
(Feel free to consider or ignore other optimizations discussed in the article.)

Problem 6: Modified Grover

a) In the notation used in lecture 19, with $|\phi\rangle = \cos \theta |\text{no}\rangle + \sin \theta |\text{yes}\rangle$ (where assuming that the search problem has $M$ solutions ($M \neq 0, M \neq N = 2^n$), define $|\text{yes}\rangle = \frac{1}{M^{1/2}} \sum_x |f(x) = 1 \rangle$ and $|\text{no}\rangle = \frac{1}{(2^n - M)^{1/2}} \sum_x |f(x) = 0 \rangle$), show that a single Grover iteration can be used to find a marked state whenever $M/N = 1/4$.

b) Consider a modification to Grover’s algorithm, where the ‘oracle’ now performs $V_\alpha |x\rangle = e^{i\alpha} |x\rangle$, if $|x\rangle$ is a target state, and $V_\alpha |x\rangle = |x\rangle$ otherwise. In addition, take $W_\alpha = (1 - e^{i\alpha}) |\phi\rangle \langle \phi| - 1 = H^\otimes n \left((1 - e^{i\alpha}) |0\rangle \langle 0| - 1\right) H^\otimes n$.

Show that if you use $G_\alpha = W_\alpha V_\alpha$ instead of the standard Grover iteration (note that $\alpha = \pi$ corresponds to the usual $G = WV$), you can choose $\alpha$ so that the algorithm finds a target state with probability 1 after one iteration, for any state with $M/N \geq 1/4$.
(Hint: find the coefficient of $|\text{no}\rangle$ in the $|\text{yes}\rangle, |\text{no}\rangle$ expansion of $G_\alpha |\phi\rangle$ and set it to zero.)

c) Combine the result of b) with the standard Grover iteration ($\alpha = \pi$) to find a modified Grover’s algorithm that finds a marked state with probability 1 with the smallest number of ‘oracle’ calls, for any given $M/N$.

In particular, for $M = 1$, what is the smallest number of calls to $V = V_\pi$ (as part of $G_\pi$) combined with a single $V_\alpha$ (as part of $G_\alpha$) to find the marked state with probability 1?