The first three problems relate to the factoring algorithm. Recall that we factored $N$ by constructing the unitary transformation $U$ which takes $U \vert a \rangle = \vert ax \ mod \ N \rangle$ for $0 \leq a < N$ and $\text{gcd}(x, N) = 1$. We found the minimum $r > 0$ for which $U^r \vert 1 \rangle = \vert 1 \rangle$ and used it to factor $N$. Note that for some of these problems, Theorem A4.10 on page 632 of N&C may come in handy. This theorem says that the multiplicative group of residues mod $p^a$ is cyclic for odd primes $p$.

1. For $N = 15$, what fraction of the residues $1 \leq x < N$ with $\text{gcd}(x, N) = 1$ will result in a factorization? How about for $N = 63$? (While testing all these residues is one way to solve this problem, there are much more efficient ones.)

2. Suppose we try to apply the factoring algorithm to a number $N = p^a$ which is a power of $p$. Will it work? If not, what goes wrong?

3. Suppose we try to apply the factoring algorithm, but we forget to check whether $\text{gcd}(x, N) = 1$ and accidentally choose an $x$ with $1 < x < N$ and $\text{gcd}(x, N) > 1$. Will the algorithm still work? If not, what goes wrong?

The next two problems deal with the period-finding algorithm on p. 236 of N&C. This was not covered in class, but is quite similar to the order-finding algorithm (p. 232) which was. The difference is that the order-finding algorithm operates on a black box $U$ which performs $U \vert a \rangle = \vert f(a) \rangle$ where $f$ is a classical one-to-one function, and finds the minimum value of $r$ such that $U^r \vert b \rangle = \vert b \rangle$, whereas the period-finding algorithm operates on a black box $U$ such that $U \vert x \rangle \vert y \rangle = \vert x \rangle \vert y \oplus f(x) \rangle$. Also note that the value of $t$ is given using big-$O$ notation, but for the algorithm to work, you actually need $t \geq 2L$.

4. Do Exercise 5.20 in N&C.

5. Suppose we apply this period-finding algorithm to the function

\[
\begin{align*}
  f(x) &= 1 \quad \text{if } r \text{ divides } x \\
  f(x) &= 0 \quad \text{if } x \text{ is not a multiple of } r.
\end{align*}
\]

Approximately what is the probability that we learn the period $r$?

6. For Grover’s search algorithm, assume that we have $M$ target states out of $N$ total states, so the black box $O$ takes

\[
\begin{align*}
  O \vert x \rangle &= -\vert x \rangle \quad \text{if } x \text{ is a target state,} \\
  O \vert x \rangle &= \vert x \rangle \quad \text{otherwise.}
\end{align*}
\]

Suppose we find a target state with probability 1 after one iteration of the algorithm. What can you say about the ratio $M/N$?
7. Consider the modification to Grover’s algorithm so that the oracle now performs

\[
O \ket{x} = e^{i\phi} \ket{x} \quad \text{if } x \text{ is a target state,}
\]
\[
O \ket{x} = \ket{x} \quad \text{otherwise.}
\]

Show that if you use the transformation

\[
\tilde{G} = H^\otimes n \left[ (1 - e^{i\phi}) \ket{0} \bra{0} - I \right] H^\otimes n O
\]

instead of the standard Grover iteration, for any state with \( M/N \) sufficiently large you can choose \( \phi \) so that the algorithm finds a target state with probability 1 after one iteration. For what values of \( M/N \) is there such a \( \phi \)?