Quantum Computation **(CMU 18-859BB, Fall 2015)**

Lecture 8: Period Finding: Simon's Problem over $\mathbb{Z}_N$

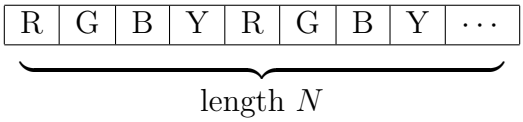
October 5, 2015

Lecturer: John Wright *Scribe: Nicolas Resch*

1 Problem

As mentioned previously, period finding is a rephrasing of Simon's algorithm, but instead of using the Fourier transform over $\mathbb{Z}_2^n$, we will use the Fourier transform over $\mathbb{Z}_N$. Moreover, an efficient quantum algorithm for period finding will "essentially" give us Shor's algorithm [Sho97] for efficient factorization – the details will be presented next lecture.

Definition 1.1 (Period-Finding Problem)**.** Given is some $f : \mathbb{Z}_N \to$ "colors" (i.e. the image of f is some "unstructured" set). Pictorially, f can be imagined as an array:



As usual, we have oracle access to f, and we denote the oracle by O_f. For this problem, it is most convenient to work with the oracle which behaves as follows:

$$O_f(|x\rangle|b\rangle) = |x\rangle|b \oplus f(x)\rangle,$$

where b is an m-qubit string. We have the promise that f is *periodic*; namely for some $s \in \mathbb{Z}_N \setminus \{0\}$, $f(x) = f(x+s)$ for all $x \in \mathbb{Z}_N$. Otherwise, all of f's values are assumed to be distinct: that is, we never have $f(x) = f(y)$ if x and y don't differ by a multiple of s. The goal of the problem is to find s.

Remark 1.2. Classically, we can actually solve this problem very efficiently. Note that the condition on s implies that s divides N. Assuming $N = 2^n$, then s must lie in the set $\{1, 2, 4, \ldots, N\}$. So we obtain an efficient classical algorithm by simply testing if $s = 1$ is f's period, then if $s = 2$ is f's period, etc. This requires us to test $n = \log N$ values of s, so the query complexity, and run-time, is $O(n)$.

So, why do we care about solving this quantumly? Shor's algorithm [Sho97] for factoring will actually look at a variant where f is "almost-periodic". So we will not necessarily know that s divides N. However, the quantum algorithm we develop today will generalize to account for this case.

2 The Algorithm

Here is the quantum algorithm that we will use to solve this problem:

- Prepare the state $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle$, i.e. the uniform superposition of all the kets $|x\rangle$.
- Attach the state $|0^m\rangle$, thereby obtaining $\left(\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle \right) \otimes |0^m\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle |0^m\rangle$.
- Query the oracle O_f on the current input. The state of the system will now be

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} |x\rangle |f(x)\rangle. \tag{1}$$

- Measure the color register, i.e. the registers corresponding to the $|0^m \oplus f(x)\rangle$ part of the output of O_f.
- Apply $\mathcal{F}_N$ to the remaining n qubits.
- Measure the remaining n qubits.
- Then, we do a little bit of "classical" computation with the output, which will be made clear later on.

To aid with the analysis of the algorithm, we will use the following notation:

Notation 2.1. *For each color c, define $f_c : \mathbb{Z}_N \to \{0, 1\}$ by*

$$f_c(x) = \begin{cases} 1 & \text{if } f(x) = c, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, f_c is the indicator function for the event that $f(x) = c$.

Thus, (1) can be rewritten as

$$\sum_{\text{colors } c} \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} (f_c(x)|x\rangle) \otimes |f(x)\rangle. \tag{2}$$

Indeed, if $f_c(x) = 0$ then the term $|x\rangle|f(x)\rangle$ disappears from the sum, so we only count $|x\rangle|f(x)\rangle$ for the one color c such that $f(x) = c$. Moreover, we comment at that in the summation (2), the number of nonzero terms of the form $(f_c(x)|x\rangle)$ is precisely $\frac{N}{s}$, which is the number of distinct values $x \in \mathbb{Z}_N$ that map to the color c under f.

3 Analysis

The first question that we should ask is: what do we get when we perform the color measurement? That is, with what probability do we measure a fixed color c? Using (2), we see that this probability is precisely

$$\begin{aligned}\sum_{x\in\mathbb{Z}_N}\left(\frac{1}{\sqrt{N}}f_c(x)\right)^2 &= \frac{1}{N}\sum_{x\in\mathbb{Z}_N} f_c(x)^2 = \frac{1}{N}\sum_{x\in\mathbb{Z}_N} f_c(x)\\ &= \mathop{\mathbf{E}}_{x\in_R\mathbb{Z}_N}[f_c(x)] = \mathop{\mathbf{Pr}}_{x\in_R\mathbb{Z}_N}[f(x)=c]\\ &= \text{fraction of } f \text{ outputs which have color } c\\ &= \frac{1}{s}.\end{aligned}$$

In the above computations, we write $x \in_R \mathbb{Z}_N$ to denote the distribution in which x is sampled from $\mathbb{Z}_N$ uniformly at random. We used the fact that $f_c(x)^2 = f_c(x)$ for all x since f_c is $\{0,1\}$-valued. We also observed that $\mathbf{E}_{x\in_R\mathbb{Z}_N}[f_c(x)] = \mathbf{Pr}_{x\in_R\mathbb{Z}_N}[f_c(x)=1]$ since $f_c(x)$ for random x is a $\{0,1\}$-valued random variable, and then the simple fact that $f_c(x)=1 \iff f(x)=c$ by the definition of f_c to conclude $\mathbf{Pr}_{x\in_R\mathbb{Z}_N}[f_c(x)=1] = \mathbf{Pr}_{x\in_R\mathbb{Z}_N}[f(x)=c]$.

Thus, we obtain a *uniformly random* color as our answer!

Given that we have observed a color c, what does the state (2) collapse to? Well, it's the substate that is consistent with the color c. This is the state

$$\left(\sum_{x\in\mathbb{Z}_N} f_c(x)|x\rangle\right)\otimes|c\rangle \text{ normalized.}$$

The normalizing factor is $\sqrt{\frac{s}{N}}$, so the state is

$$\sqrt{\frac{s}{N}}\left(\sum_{x\in\mathbb{Z}_N} f_c(x)|x\rangle\right)\otimes|c\rangle$$

It will be most convenient for us to rewrite this state as

$$\frac{1}{\sqrt{N}}\sum_{x\in\mathbb{Z}_N}\sqrt{s}f_c(x)|x\rangle. \tag{3}$$

At this point, we will use the "powerful trick" that always seems to work when we analyze quantum algorithms. This trick is, of course, the quantum Fourier transform. More precisely, we will:

Apply the Quantum Fourier Tranform over $\mathbb{Z}_N$ and measure.

Let us briefly recall what generally happens when we use this trick. If we have a function $g : G \to \mathbb{C}$ such that $\mathbf{E}_{x\in_R G}[|g(x)|^2] = 1$, where G is either $\mathbb{Z}_2^n$ or $\mathbb{Z}_N$, then the following is a valid quantum state:

$$\frac{1}{\sqrt{N}}\sum_{x\in G} g(x)|x\rangle.$$

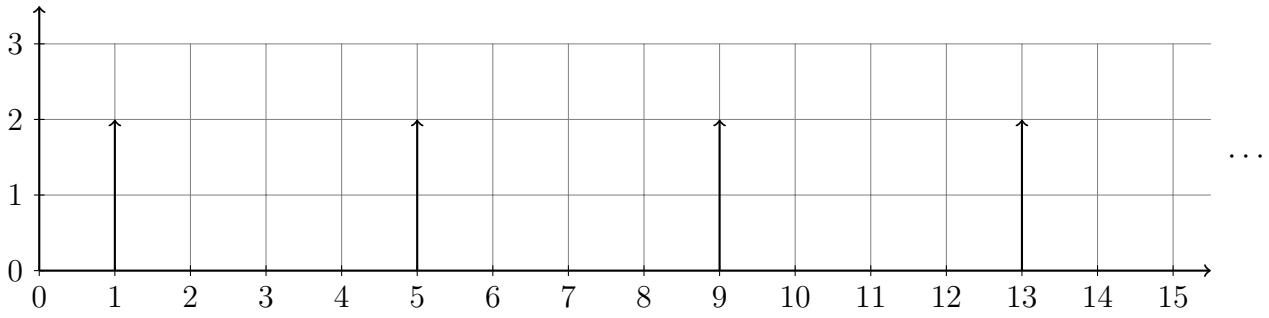


Figure 1: The function $g = 2 \cdot 1_{\{1,5,9,13,\ldots,\}}$

If we apply the Fourier transform, which corresponds to applying H_N if $G = \mathbb{Z}_2^n$ and $\mathcal{F}_N$ if $G = \mathbb{Z}_N$, then the state we obtain is

$$\sum_{\gamma \in G} \hat{g}(\gamma)|\gamma\rangle.$$

Thus, when we measure, we observe $\gamma \in G$ with probability $|\hat{g}(\gamma)|^2$.

Remark 3.1. This procedure is called *spectral sampling*. The set $\{\hat{g}(\gamma)\}_{\gamma \in G}$ is called the *Fourier spectrum* of g. This is *almost* always how exponential speed-ups are obtained in quantum computing.

In our case, recalling (3), we should put $g = \sqrt{s} \cdot f_c$, as then $\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} g(x)|x\rangle$ is the quantum state after measuring the color register.

Remark 3.2. For Simon's algorithm, if we define

$$1_y(x) = \begin{cases} 1 & x = y, \\ 0 & x \neq y \end{cases},$$

we had $g = \sqrt{2}(1_y + 1_{y+s})$, where $f^{-1}(c) = \{y, y+s\}$ for some color c.

Before continuing, let's define the following notation which we will use throughout the analysis:

Notation 3.3. *For $S \subseteq \mathbb{Z}_N$, we define $1_S : \mathbb{Z}_N \to \{0,1\}$ by*

$$1_S(x) = \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{if } x \notin S. \end{cases}$$

Example 3.4. *Say $s = 4$, c is Green, and f is Green on $1, 5, 9, 13, \ldots$. Then $g = \sqrt{s} \cdot f_c = 2 \cdot 1_{\{1,5,9,13,\ldots\}}$. Figure 1 demonstrates what this function looks like.*

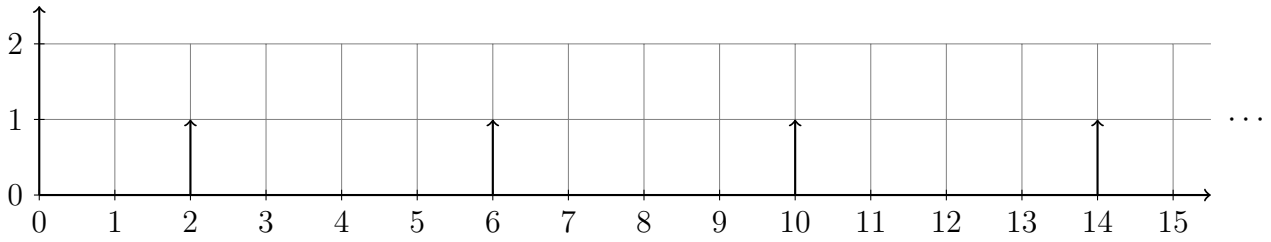


Figure 2: The function $f_{\text{Green}} = 1_{\{2,6,10,\ldots\}}$

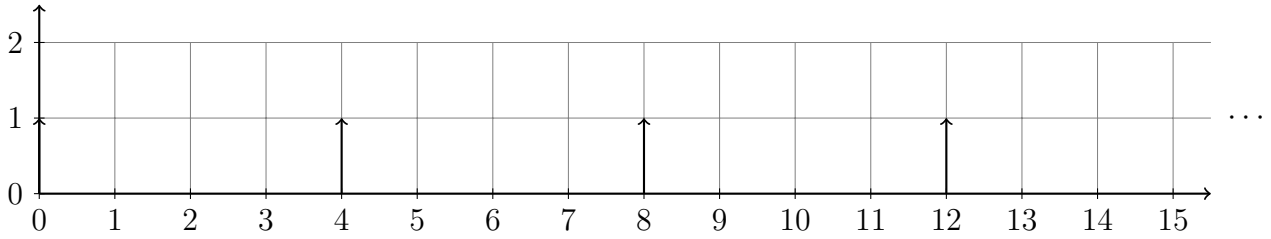


Figure 3: The function $f_{\text{Red}} = 1_{\{0,4,8,\ldots\}}$

Our algorithm outputs $\gamma \in \mathbb{Z}_N$ with probability

$$|\hat{g}(\gamma)|^2 = s \cdot |f_c(\gamma)|^2.$$

So, the question now is: *what are these Fourier coefficients?* We have a periodic spike function. It turns out that the Fourier transform of such a function is also a periodic spike function.

To simplify our analysis, we would like to show that $|f_c(\gamma)|^2$ is independent of c. That way, it will suffice to compute $|f_c(\gamma)|^2$ for one "nice" choice of c. Hence, we will prove the following claim:

Claim 3.5. *Let $g : \mathbb{Z}_N \to \mathbb{C}$ and let $t \in \mathbb{Z}_N$. Define $g^{+t} : \mathbb{Z}_N \to \mathbb{C}$ by*

$$g^{+t}(x) = g(x + t).$$

Then g and g^{+t} have "essentially the same" Fourier coefficients. That is, they differ by a multiplicative factor of magnitude 1, so they have the same magnitude.

Why is this helpful? Well, say $f_{\text{Green}} = 1_{\{2,6,10,\ldots,\}}$ and $f_{\text{Red}} = 1_{\{0,4,8,\ldots,\}}$. Then, these two functions are shifts of each other, as $f_{\text{Green}} = f_{\text{Red}}^{+2}$. It therefore suffices to study f_c for a single value of c, as desired. See figures 2 and 3.

Proof. Let $\omega = e^{\frac{2\pi i}{N}}$, so that $\chi_\gamma(x) = \omega^{\gamma \cdot x}$. We compute:

$$\begin{aligned}
\widehat{g^{+t}}(\gamma) &= \mathop{\mathbf{E}}_{x \in_R \mathbb{Z}_N}[g^{+t}(x)\chi_\gamma(x)^*] \\
&= \mathop{\mathbf{E}}_{x \in_R \mathbb{Z}_N}[g(x+t)\chi_\gamma(x)^*] \qquad (*)
\end{aligned}$$

At this point, we make the change of variables $y = x + t$. For fixed t, if x is selected from $\mathbb{Z}_N$ uniformly at random, so is y. Thus,

$$\begin{aligned}(*) &= \mathop{\mathbf{E}}_{y \in_R \mathbb{Z}_N}[g(y)\chi_\gamma(y-t)^*] \\ &= \mathop{\mathbf{E}}_{x \in_R \mathbb{Z}_N}[g(y)\chi_\gamma(y)^*\chi_\gamma(-t)^*] \\ &= \chi_\gamma(-t)^* \mathop{\mathbf{E}}_{y \in \mathbb{Z}_N}[g(y)\chi_\gamma(y)^*] \\ &= \omega^{\gamma t}\hat{g}(\gamma).\end{aligned}$$

Recalling that $\omega^{\gamma t}$ is an N-th root of 1, we conclude that it is a complex number of magnitude 1. In the above computations, we used the important fact that $\chi_\gamma(x+y) = \chi_\gamma(x)\chi_\gamma(y)$ for all $x, y \in \mathbb{Z}_N$, as well as the observation that $\chi_\gamma(-t)^*$ does not depend on the randomly selected $y \in_R \mathbb{Z}_N$. □

This immediately yields the following corollary:

Corollary 3.6. $|\widehat{g^{+t}}(\gamma)|^2 = |\omega^{\gamma t}|^2|\hat{g}(\gamma)|^2 = |\hat{g}(\gamma)|^2$.

Thus, the probability of sampling $\gamma \in \mathbb{Z}_N$ is independent of t. In our case, this means that when we do the spectral sampling at the end of the algorithm, it does not matter what color we measured earlier. It is therefore no loss of generality to assume that, if c is the color we sampled, $f(0) = c$, from whence it follows that $f_c = 1_{\{0,s,2s,3s,\ldots\}}$.

What is so special about the set $\{0, s, 2s, \ldots\}$? It's precisely the subgroup of $\mathbb{Z}_N$ generated by the number s!

Remark 3.7. For Simon's algorithm, we would study $1_{\{0,s\}}$, as $\{0, s\}$ is the subgroup of $\mathbb{Z}_2^n$ generated by s.

We are now prepared to analyze the Fourier coefficients of g.

Proposition 3.8. *Let $H = \{0, s, 2s, \ldots\} \subseteq \mathbb{Z}_N$ and let $h = 1_H$. Then*

$$\hat{h}(\gamma) = \begin{cases} \frac{1}{s} & \text{if } \gamma \in \{0, \frac{N}{s}, \frac{2N}{s}, \frac{3N}{s}, \ldots\}, \\ 0 & \text{otherwise.} \end{cases}$$

Remark 3.9. Observe that $|\{0, \frac{N}{s}, \frac{2N}{s}, \frac{3N}{s}, \ldots\}| = s$. To see this, recall that $s \cdot \frac{N}{s} = N = 0 \mod N$, so the size of the set is at most s because $\frac{aN}{s} = \frac{(a+s)N}{s}$ for all values of a. But if $r < s$ then $r \cdot \frac{N}{s} \neq 0 \mod N$, so if $a, b < s$ with $a < b$ then $\frac{aN}{s} \neq \frac{bN}{s}$ as otherwise $\frac{(b-a)N}{s} = 0 \mod N$ even though $b - a \leq b < s$. Thus, we conclude that there are precisely s values of γ for which $\hat{h}(\gamma) \neq 0$.

Assuming the proposition, what do we conclude? Well, the state changes as follows. Recalling that the state prior to applying the $\mathcal{F}_N$ gate is (3), we have

$$\frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} \sqrt{s} \cdot h(x)|x\rangle \xrightarrow{\mathcal{F}_N} \sum_{\gamma \in \mathbb{Z}_N} \sqrt{s} \cdot \hat{h}(\gamma)|\gamma\rangle$$
$$\xrightarrow{\text{measure}} \gamma \text{ with probability } s \cdot |\hat{h}(\gamma)|^2.$$

Since $\hat{h}(\gamma) = \frac{1}{s}$ iff $\gamma \in H$, we measure $\gamma \notin H$ with probability 0 and $\gamma \in H$ with probability $s \cdot \left|\frac{1}{s}\right|^2 = \frac{1}{s}$. That is, we sample a uniformly random $\gamma \in H$. Recalling that $H = \{0, \frac{N}{s}, \frac{2N}{s}, \ldots\}$, any $\gamma \in H$ satisfies $\gamma s = 0 \mod N$. Thus, we conclude that we sample a uniformly random γ such that $\gamma s = 0 \mod N$. This is just like what happened in Simon's algorithm! There, we sampled a uniformly random $\gamma \in \mathbb{Z}_2^n$ such that $\gamma \cdot s = 0$, so the only difference is that now we consider multiplication modulo N instead of the dot product of two length n strings modulo 2.

Now, we will prove Proposition 3.8.

Proof. We compute

$$\hat{h}(\gamma) = \mathop{\mathbf{E}}_{x \in_R \mathbb{Z}_N}[h(x) \cdot \chi_\gamma(x)^*] = \frac{1}{s} \mathop{\mathbf{E}}_{x \in_R H}[\chi_\gamma(x)^*] = (\dagger).$$

The second equality follows from the fact that $h(x)$ is only ever nonzero on the set H which has size s. We consider two cases:

Case 1. Suppose $\gamma \in \{0, \frac{N}{s}, \frac{2N}{s}, \ldots\}$. Then

$$\chi_\gamma(x)^* = \omega^{-\gamma \cdot x}$$

Now, recall that x is a multiple of s, since s is assumed to be sampled from H. Similarly, if γ is in $\{0, \frac{N}{s}, \frac{2N}{s}, \ldots\}$, γ is a multiple of $\frac{N}{s}$. Thus,

$$\chi_\gamma(x)^* = \omega^{-(\text{multiple of } \frac{N}{s}) \cdot (\text{multiple of } s)} = \omega^{-(\text{multiple of } N)} = 1,$$

using that ω is an N-th root of unity. Thus, $\chi_\gamma(x)^* = 1$ for all $x \in H$, so

$$(\dagger) = \frac{1}{s} \cdot \mathop{\mathbf{E}}_{x \in H}[1] = \frac{1}{s}.$$

Case 2. We could, using some elementary arithmetic, directly compute $\mathbf{E}_{x \in_R H}[\chi_\gamma(x)^*]$ to show that it is 0. However, we'll be a bit more slick. We've already shown that our algorithm outputs $\gamma \in \{0, \frac{N}{s}, \frac{2N}{s}\}$ with probability $s \cdot \left(\frac{1}{s}\right)^2 = \frac{1}{s}$. Since $|\{0, \frac{N}{s}, \frac{2N}{s}, \ldots, \}| = s$, there is no probability left to give to the $\bar{h}(\gamma)$'s! That is,

$$1 = \sum_{\gamma \in \mathbb{Z}_N} s|\hat{h}(\gamma)|^2 = \sum_{\gamma \in H} s|\hat{h}(\gamma)|^2 + \sum_{\gamma \notin H} s|\hat{h}(\gamma)|^2 = 1 + \sum_{\gamma \notin H} s|\hat{h}(\gamma)|^2$$

so

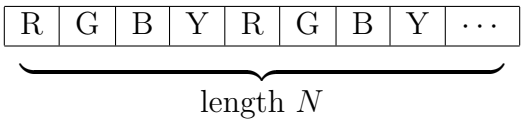
$$\sum_{\gamma \notin H} s|\hat{h}(\gamma)|^2 = 0,$$

implying $\hat{h}(\gamma) = 0$ for all $\gamma \notin H$.

□

4 Summary

Let's reviewed what we've accomplished. We know that f is promised to be of the form



That is, f is s-periodic. We have shown that, with one query to O_f and $O(n^2)$ gates along with a couple measurements, we get a uniformly random $\gamma \in \mathbb{Z}_N$ such that $\gamma \cdot s = 0 \mod N$ (or, equivalently, such that $\gamma \in \{0, \frac{N}{s}, \frac{2N}{s}, \ldots\}$).

Well, what should we do? As is usually the case, we'll want to repeat the algorithm some number of times. But how many times show we repeat the algorithm to find s?

First of all, we observe that to find s, it actually suffices to find $\frac{N}{s}$. After that, we can divide through by N and take the reciprocal to compute s.

So, how do we find $\frac{N}{s}$? For the time being, forget that N and s are powers of 2. Let $m = \frac{N}{s}$, so our algorithm samples a random integer multiple of m. Suppose we have two random samples, which we can write am and bm. Note that

$$\gcd(am, bm) = \gcd(a, b)m.$$

So, if $\gcd(a, b) = 1$, by taking the gcd of the two outputs am and bm, we will have found m! Thus, the question becomes the following: If a and b are sampled from $\{0, 1, \ldots, s-1\}$ uniformly at random and independently, what's the probability that $\gcd(a, b) = 1$?

Claim 4.1. *If a and b are sampled as above, $\mathbf{Pr}[\gcd(a, b) = 1] \geq \Omega(1)$.*

Proof. First condition on the event that a and b are not equal to zero. This event occurs with large probability: $\left(\frac{s-1}{s}\right)^2$ which is at least $1/4$. Fix a prime p. Observe that

$$\begin{aligned}
\mathbf{Pr}[p \text{ divides } a \text{ and } b] &= \mathbf{Pr}[p \text{ divides } a] \cdot \mathbf{Pr}[p \text{ divides } b] && \text{since } a \text{ and } b \text{ are independent} \\
&= \mathbf{Pr}[p \text{ divides } a]^2 && \text{since } a \text{ and } b \text{ are identically distributed}
\end{aligned}$$

Note that at most a $1/p$ fraction of the elements in the set $\{1, 2, \ldots, s-1\}$ are multiples of p. Since we are conditioning on the event that $a \neq 0$, we conclude that $\mathbf{Pr}[p \text{ divides } a] \leq 1/p$. Therefore

$$\mathbf{Pr}[p \text{ divides } a \text{ and } b] \leq \frac{1}{p^2}.$$

Thus,

$$\begin{aligned}
\mathbf{Pr}[\gcd(a, b) > 1] &= \mathbf{Pr}[a \text{ and } b \text{ share a prime factor}] \\
&\leq \sum_{\text{primes } p} \frac{1}{p^2} && \text{Union Bound} \\
&\leq \sum_{n \geq 2} \frac{1}{n^2} = \frac{\pi^2}{6} - 1 \approx 0.6,
\end{aligned}$$

where we have used the famous number-theoretic fact that

$$\sum_{n \geq 1} \frac{1}{n^2} = \zeta(2) = \frac{\pi^2}{6},$$

where ζ denotes the Riemann zeta function. □

Remark 4.2. Computing $\sum_{\text{primes } p} \frac{1}{p^2}$ exactly is an open problem.

References

[Sho97] Peter Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM journal on computing*, 26(5):1484–1509, 1997.