1 Prime Number Generation

1.1 Uniform prime generation

The original 3-step algorithm [cite], in essence, generates two random prime numbers that are distributed uniformly in a given interval. Ahlswede then uses the fact that the primes are uniform to bound the second kind error probability. [The relaxation of this assumption complicates the error analysis. Explore this in subsection x.].

To generate uniform primes on the interval $]1, \pi_K]$, Ahlswede picks a random index $k \leftarrow \{1, \ldots, K\}$ and then calculates π_k. Although this method minimizes the number random bits and generates exactly uniform primes, it is computationally expensive with state-of-the-art algorithms compute π_k in $2^{\mathcal{O}(K)}$[cite].

To improve the time complexity of the scheme, we look for polynomial time algorithms that produce almost uniform primes with the least number of random bits possible. A trivial algorithms for producing uniform primes is given in Algorithm 1.

```
Algorithm 1: Generating uniform primes
   input  : Positive integer $n$.
   output: A uniformly choen prime number less than or equal to $n$.
 1 repeat
 2     $p \leftarrow \{2, \ldots, n\}$;
 3 until $p$ is prime
 4 return $p$
```

When we use a deterministic primality test in line 3 of Algorithm 1 the distribution of primes is exactly uniform. This algorithm may never terminate, however, we expect it to stop after $\mathcal{O}(\lg n)$ steps. Because, by the prime number theorem [reference],

$$\frac{\pi(n)}{n} \approx \frac{1}{\ln n} \tag{1}$$

hence, on average after $\mathcal{O}(\lg n)$ steps a prime number p is chosen. As a result, Algorithm 1 uses an average of $(\lg n)^2$ random bits. The current state-of-the-art deterministic primality tests, AKS, runs in $\tilde{\mathcal{O}}\Big((\lg n)^6\Big)$ which means that on average Algorithm 1 runs in $\tilde{\mathcal{O}}\Big((\lg n)^7\Big)$ [cite].

We can further improve the time complexity of the Algorithm 1 if we use randomized primality tests. These tests can determine whether a number p is prime with high probability.

```
Algorithm 2: Generating uniform primes
   input  : Positive integer $n$.
   output: A uniformly choen prime number less than or equal to $n$.
 1 repeat
 2     $p \leftarrow \{2, \ldots, n\}$;
 3 until $p$ is probably a prime
 4 return $p$
```

For example, the Miller-Rabin test might declare a composite number as a prime, however, this happens with low probability, as low as desired. The output of the Algorithm 2 is not a unifrom prime number as it can be composite, however, the distribution of prime numbers is equiprobable over all primes. Continuing with Miller-Rabin test, it uses $\lg p$ random bits where p is the number that is to be tested. Therefore, we still use an average of $2(\lg n)^2$ random bits. The test itself runs in $\mathcal{O}\Big((\lg n)^3\Big)$ hence, the Algorithm 2 runs in $\mathcal{O}\Big((\lg n)^4\Big)$.

In this report, we implement the Miller-Rabin test since it is more efficient and easier to implement. Furthermore, by exectuing this test an appropriate number rounds, we can ensure that the resulting distribution is statistically close to the uniform distribution over primes.

1.2 Miller-Rabin analysis

Miller-Rabin is a well-known random primality test algorithm. Let $MR(n, k)$ be the distribution of Miller-Rabin algorithm on the prime candidate n with k repeats. Let $\mathcal{P}$ be the set of primes. Then, from Theorem 31.39 of [2] we have the following probabilities.

$$
\begin{aligned}
\Pr[MR(n,k) \mid n \in \mathcal{P}] &= 1 \qquad &(2)\\
\Pr[MR(n,k) \mid n \notin \mathcal{P}] &\leq 4^{-k} \qquad &(3)
\end{aligned}
$$

Consider the following random prime number genrator, $GMR(N, s, k)$. This algorithm, samples numbers uniformly and then checks if they are prime using the Miller-Rabin test. The parameter N is the upperbound, s is the maximum number of attempts, and k is the number of repeats in the underlying Miller-Rabin test. We analyse the distribution of $GMR(N, s, k)$. Let n_i denote the random variable n in the i_{th} iteration.

```
Algorithm 3: $GMR(N, s, k)$
input  : positive integers $N, s, k$
output: A uniformly chosen prime number less than or equal to $N$
for $i = 1 \to s$ do
    $n \leftarrow \{1, 2, \ldots, N\}$
    if $MR(n, k)$ then
        return $n$
    end
end
return $\perp$
```

$$
\begin{aligned}
\Pr[GMR(M, s, k) = \perp] &= \Pr[MR(n_1, k) = \cdots = MR(n_s, k) = 0] && (4)\\
&= \prod_{i=1}^{s} \Pr[MR(n_i, k) = 0] \qquad \text{(Independence)} && (5)\\
&= \prod_{i=1}^{s} \Pr[MR(n_i, k) = 0 \mid n_i \notin P] \Pr[n_i \notin P] && (6)\\
&\leq \prod_{i=1}^{s} \left(1 - \frac{\pi(N)}{N}\right) && (7)\\
&= \left(1 - \frac{\pi(N)}{N}\right)^{s} && (8)\\
&\approx \left(1 - \frac{1}{\ln N}\right)^{s} \qquad \text{(PNT)} && (9)
\end{aligned}
$$

If we bound this error probability with ϵ, then we get the following bound on s.

$$
s \geq \frac{\ln \epsilon}{\ln\left(1 - \frac{1}{\ln N}\right)} \approx -\ln N \ln \epsilon
$$

for sufficiently large N.

The probability that the result of $GMR(N, s, k)$ is composite, given it is not $\perp$ is as follows.

$$
\begin{aligned}
\Pr[GMR(N, s, k) \notin P \mid GMR(N, s, k) \neq \perp] &\leq \sum_{i=1}^{s} \Pr[MR(n_i, k) = 1, n_i \notin P] && (10)\\
&= \sum_{i=1}^{s} \Pr[MR(n_i, k) = 1 \mid n_i \notin P] \Pr[n_i \notin P] && (11)\\
&= \sum_{i=1}^{s} 4^{-k} \left(1 - \frac{\pi(N)}{N}\right) && (12)\\
&= s4^{-k} \left(1 - \frac{\pi(N)}{N}\right) && (13)\\
&\approx s4^{-k} \left(1 - \frac{1}{\ln N}\right) \qquad \text{(PNT)} && (14)
\end{aligned}
$$

If we bound this error probability with δ, then we get the following bound on s.

$$s \leq \left(1 - \frac{1}{\ln N}\right)^{-1} 4^k \delta \approx 4^k \delta \tag{15}$$

Let $\epsilon = e^{-l}$ and $\delta = 2^{-q}$ with $l, q \geq 0$. Then,

$$l \ln N \leq s \leq 2^{2k-q} \tag{16}$$

Note that, $s = l \ln N$ and $k = \dfrac{\lg(l/e) + \lg\lg N + q}{2}$ satisfies both inequalities.

1.3 Prime Number Theorem

Theorem 1 ([1]). *Let $\pi(x)$ denote the number of primes less than or equal to $x \geq 1$. The prime number theorem states that*

$$\lim_{x\to\infty} \frac{\pi(x)\ln x}{x} = 1$$

That is, $\pi(x) \sim \dfrac{x}{\ln x}$. This implies that n-th prime p_n is asymptotically given by $p_n \sim n \ln n$.

Moreover, we may frequently use these exact bounds on $\pi(x)$.

Lemma 2 ([1]). *For any $n \geq 2$,*

$$\frac{1}{6}\frac{n}{\ln n} \leq \pi(n) \leq 6\frac{n}{\ln n}$$

and for any $n \geq 1$,

$$\frac{1}{6} n \ln n \leq p_n \leq 12(n \ln n + n \ln \frac{12}{e})$$

References

[1] Tom M. Apostol. *Introduction to Analytic Number Theory.* Springer New York, 1976.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition.* The MIT Press, 3rd edition, 2009.