# 1 Kurepa's Conjecture

Kurepa conjectured that

$$\sum_{k=0}^{p-1} k! \not\equiv 0 \bmod p$$

for all odd primes $p$.

# 2 Notation

First we define notation that will be used throughout this text.

We will use $\log(x)$ to denote the natural logarithm of $x$ and $\lg(x)$ to denote the logarithm of $x$ to base 2.

Let $f$ and $g$ be functions from $\mathbb{R}$ to $\mathbb{R}$. We say that $f(x) = O(g(x))$ if $|f(x)| \leq M|g(x)|$ for all $x > x_0$ for some positive constants $M$ and $x_0$. This is referred to in the literature as big $O$ notation or Landau notation.

We use $\beta(X)$ to denote the number of bits required to store $X$ in memory and includes a variable $C$ to estimate the effects of overheads. For example, if $X$ is a positive integer $n$, then $\beta(X) = \lg(n) + C$ where the $C$ accounts for overhead. For the matrix $X = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, we define $\beta(X)$ as

$$\begin{aligned} \beta(X) &= \max(\lg(a) + \lg(c), \lg(b) + \lg(d)) + C \\ &= \|X\|_1 + C \end{aligned}$$

where $\|\cdot\|_1$ denotes the operator norm for $p = 1$.

We assume that $\beta$ is multiplicative; if $X$ and $Y$ are objects where $X * Y$ is defined, then $\beta(X * Y) = \beta(X) + \beta(Y)$.

We define the $n$-th moduli $m_n = \begin{cases} n & n \text{ is prime} \\ 1 & \text{otherwise} \end{cases}$, where $n$ is a positive integer. Hence $\beta(m_n) = \begin{cases} \lg(n) + C & n \text{ is prime} \\ C & \text{otherwise} \end{cases}$.

Furthermore we define the $n$-th object $A_n = \begin{pmatrix} n & 0 \\ n & 1 \end{pmatrix}$ where $n$ is a positive integer. It follows that $\beta(A_n) = 2\lg(n) + C$.

# 3 Memory Usage

In this section we will calculate the space required to implement an accumulating remainder tree to test Kurepa's conjecture. First we consider the naive case from 0 to $N$ and estimate the necessary space. Then we calculate the space required if we implement Stage 1 and Stage 2 without a remainder forest for Stage 2. Finally we estimate the space required if we implement Stage 1 and Stage 2 with a remainder forest for Stage 2.

### 3.1 Naive Method

First we consider the space required to compute the moduli product tree for the interval $[1, N]$. The number of bits required to store the bottom level of the tree is $\sum_{n=1}^{N} \beta(m_n)$. Now the number of bits to store the second lowest level is at most $\sum_{n=1}^{N/2} \beta(m_{2n-1}m_{2n}) = \sum_{n=1}^{N} \beta(m_n)$ due to the submultiplicativity of $\beta$. Likewise, every level of this tree requires $\sum_{n=1}^{N} \beta(m_n)$ bits and the number of levels is $\lg(N)$ so the total size required is at most

$$\begin{aligned}\lg(N) \sum_{n=1}^{N} \beta(m_n) &= \lg(N) \left( \sum_{\substack{p \ prime \\ p \le N}} \lg(p) + \sum_{k=1}^{N} C \right) \\ &= \lg(N) \left( \Theta(N)/\log(2) + CN \right)\end{aligned}$$

where $\Theta(N)$ is the first Chebyshev function.

Now we consider the space required to compute the object product tree for the same interval $[1, N]$. As above, the number of bits to store each level is the same and there are $\lg(N)$ levels. The space taken for the lowest level is

$$\begin{aligned}\sum_{n=1}^{N} \beta(A_n) &= \sum_{n=1}^{N} 2\lg(N) + C \\ &= 2\lg(N!) + CN \\ &\approx CN + 2N\left(\lg(N) - \lg(e)\right) \\ &= N\left(2\lg(N) + C - 2\lg(e)\right)\end{aligned}$$

where the approximation is found by using Stirling's approximation formula. Hence the total space required for the object product tree is $N\lg(N)\left(2\lg(N) + C - 2\lg(e)\right)$.

For the remainder tree, only one level of the tree is stored at any one time and each entry is bounded by the moduli so the space required is $2\Theta(N)/\log(2) + C * N$. Hence the total size of the trees in bits is

$$N\lg(N)\left(2\lg(N) + C - 2\lg(e)\right) + \lg(N)\left(\Theta(N)/\log(2) + CN\right) + 2\Theta(N)/\log(2) + CN.$$

Note that $\Theta$ has not been approximated.

### 3.2 Stage 1 and Stage 2

The aim of Stage 1 is to calculate $V := A_{m-1} \cdots A_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \bmod m_M \cdots m_{N-1}$. To calculate this, $m_M \cdots m_{N-1}$ is calculated using a product tree and then $A_{m-1} \cdots A_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ is calculated modulo $m_M \cdots m_{N-1}$ in blocks of matrix products such that the bit size of the block is at most $\beta(m_M \cdots m_{N-1})$ . The bits

required to store $m_M \cdots m_{N-1}$ is

$$\begin{aligned}\beta(m_M \cdots m_{N-1}) &= \log(\prod_{M \le n < N} m_n) + C \\ &= C + \sum_{\substack{M \le p < N \\ p \ prime}} \lg(p) \\ &= C + (\Theta(N-1) - \Theta(M))/\log(2)\end{aligned}$$

. Hence $\beta(V) = 2(\Theta(N-1)-\Theta(M))/\log(2)+C$. Now during the multiplication, the product of matrices requires $2(\Theta(N-1) - \Theta(M))/\log(2) + C$ bits and the total size of Stage 1 is

$$5(\Theta(N-1) - \Theta(M))/\log(2) + 3C$$

bits. Now

$$\begin{aligned}5(\Theta(N-1) - \Theta(M))/\log(2) + 3C &\approx 5(1.001093(N-1) - 0.998697M)/\log(2) + 3C \\ &\approx 5(N-M)/\log(2) + 3C\end{aligned}$$

where the approximations are true if $M \ge 1155901$. NEED TO REFERENCE "Sharper bounds for the chebyshev Functions II" by Schoenfeld.

Now we calculate the space required for Stage 2 without using a remainder forest. This means full trees are required for the interval $[M, N)$.

Now we use a remainder forest for Stage 2. Suppose that $[M, N)$ is subdivided into intervals of width $w$. Now of all the trees in the remainder forest, the first one takes the most space as it has the largest moduli so considering the first tree will provide an upper bound of the bits required.

The initial vector has size $2(\Theta(N-1) - \Theta(M))/\log(2) + C$. The space required for the moduli product tree is $\lg(w)\,(\Theta(M+w) - \Theta(M)) + Cw$ and the space required for the matrix tree is $\lg(w)\,(Cw + 2\lg(w)\sum_{n=0}^{w}\lg(M+n))$. The size of the remainder tree is $2\,(\Theta(M+w) - \Theta(M))\,/\log(2) + Cw$. Thus

the total size is

$$
\begin{aligned}
&2(\Theta(N-1)-\Theta(M))/\log(2)+C+\lg(w)\left(\Theta(M+w)-\Theta(M)\right)+Cw\\
&+\lg(w)\left(Cw+2\lg(w)\sum_{n=0}^{w}\lg(M+n)\right)+2\left(\Theta(M+w)-\Theta(M)\right)/\log(2)+Cw\\
\approx&2(N-M)\log(2)+C+w\lg(w)+Cw+2w\lg(w)+Cw\\
&+\lg(w)\left(Cw+2\lg(w)\sum_{n=0}^{w}\lg(M+n)\right)\\
\approx&2(N-M)\log(2)+w\lg(w)+Cw+2w\lg(w)+Cw\\
&+\lg(w)\left(Cw+2\lg(w)\sum_{n=0}^{w}\lg(M+n)\right)\\
=&2(N-M)\log(2)+3w\lg(w)+2Cw+\lg(w)\left(Cw+2\lg(w)\sum_{n=0}^{w}\lg(M+n)\right)\\
\leq&2(N-M)\log(2)+3w\lg(w)+2Cw+\lg(w)\left(Cw+2\lg(w)\sum_{n=0}^{w}\lg(N)\right)\\
=&2(N-M)\log(2)+3w\lg(w)+2Cw+\lg(w)\left(Cw+2w\lg(w)\lg(N)\right)\\
=&(N-M)\log(2)+2w\left(\lg^2(w)\lg(N)+\lg(w)(3+C)+2C\right).
\end{aligned}
$$