PRIME NUMBERS GENERATION USING THE WHEEL SIEVE

Presented by :

Peyman Navidi

WHEEL SIEVE

Wheel factorization is a graphical method for manually performing a preliminary to the Sieve of Eratosthenes that separates prime numbers from composites.

Start by writing the natural numbers around circles. Prime numbers in the innermost circle have their multiples in similar positions as themselves in the other circles, forming spokes of primes and their multiples. Multiples of the prime numbers in the innermost circle form spokes of composite numbers in the outer circles.

PROCEDURE

- Find the first few prime numbers.
- Multiply the prime numbers together to give the result **n**.
- Write **1** to **n** in a circle. This will be the inner-most circle.
- Taking **x** to be the number of circles written so far, continue to write **xn + 1** to **xn + n** in another circle around the inner-most circle, such that **xn + 1** is in the same position as **(x − 1)n + 1**.
- Repeat step **4** until the largest number to be tested for primality.
- Strike off the number **1**.
- Strike off the spokes of prime numbers (found in step **1**) with its multiples without striking off the numbers in the inner-most circles.
- Strike off the spokes of all multiples of prime numbers found in step **1**.
- The remaining numbers in the wheel contain mostly prime numbers. Use other methods such as Sieve of Eratosthenes to remove the remaining non-primes.

EXAMPLE

1. Find the first 2 prime numbers — 2 and 3.
2. **n** = 2 × 3 = **6** wheel factorization
3. **Ring0** = 1 2 3 4 5 6
4. **X** = **1** → **xn + 1** = 1 × 6 + 1 = **7**; **(x + 1)n** = (1 + 1) × 6 = **12**; Write **7** to **12**

Ring0 = 1 2 3 4 5 6
Ring1 = 7 8 9 10 11 12

5. **X** = **2** → **xn + 1** = 2 × 6 + 1 = **13**; **(x + 1)n** = (2 + 1) × 6 = **18**; Write **13** to **18**

Ring0 = 1 2 3 4 5 6
Ring1 = 7 8 9 10 11 12
Ring2 = 13 14 15 16 17 18

EXAMPLE

6. Sieving

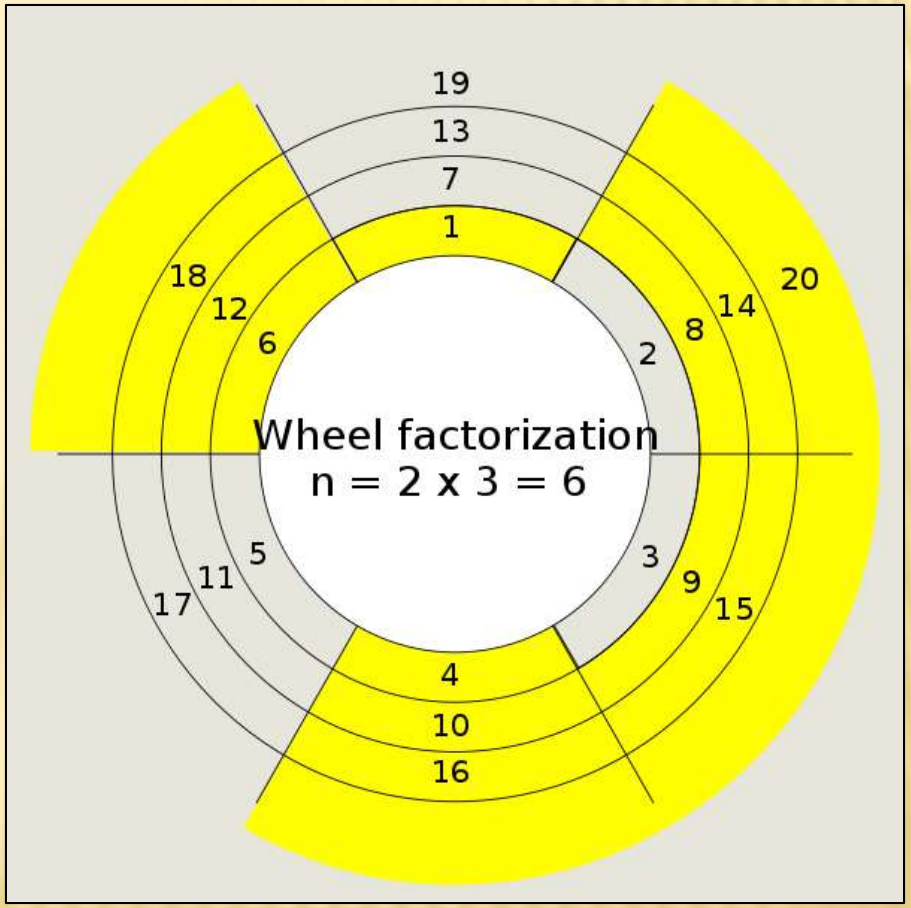
Ring0 = ~~1~~ 2 3 ~~4~~ 5 ~~6~~

Ring1 = 7 ~~8~~ ~~9~~ ~~10~~ 11 ~~12~~

Ring2 = 13 ~~14~~ ~~15~~ ~~16~~ 17 ~~18~~

Ring3 = 19 ~~20~~ ~~21~~ ~~22~~ 23 ~~24~~

Ring4 = 25 ~~26~~ ~~27~~ ~~28~~ 29 ~~30~~



7. Sieving

2 3 5 7 11 13 17 19 23 25 29

EXAMPLE

Wheel Sieve with :

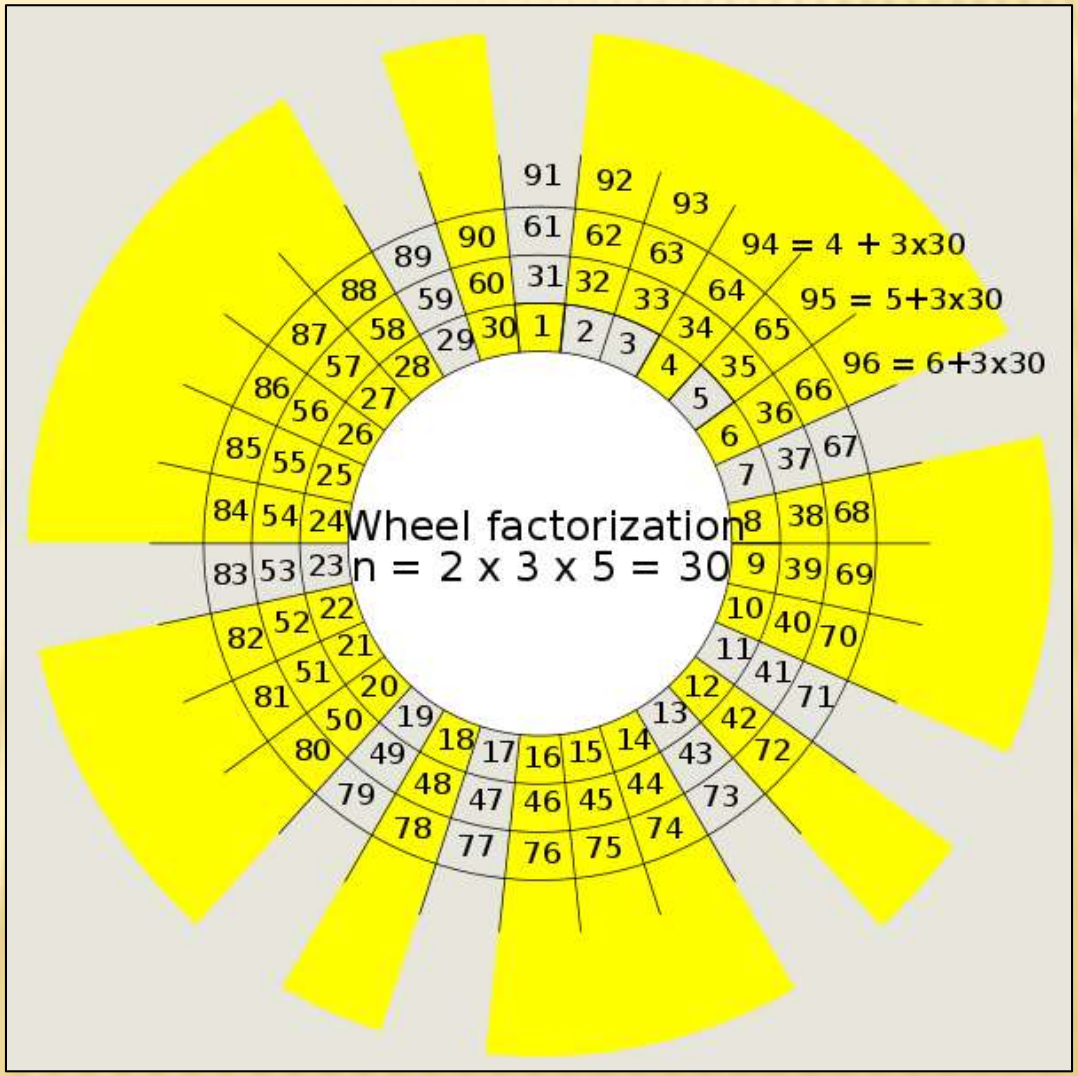
n = 2 × 3 × 5 = 30

Each number :

= pattern + (ring × n)

Order :

= O(n / log log n)



PARALLEL ALGORITHM

```
If (rank == 0)
{
    Generate wheel pattern based on wheel factorization
    Broadcast pattern to all process
}

Each process run :

    for( j = start ; j <= end ; j++ )
    if (( pattern[ k ] + rank * ring) == j )
        prime = j;
```

PARALLEL ALGORITHM

If maximum number larger than number of process :

Process repeat for each iteration :

```
for( i = 0 ; i < iteration ; i++)
{
    start = rank * ring +1 + i * proces * ring;
    end = (rank + 1) * ring + i * proces * ring;

    for( j = start ; j <= end ; j++ )
    if (( pattern[ k ] + rank * ring) == j )
        prime = j;
}
```

PARALLEL ALGORITHM

This algorithm maybe generate the series of square number. Improved this algorithm to the below :

```
for( i = 0 ; i < iteration ; i++)
    {
        ...
        for( j = start ; j <= end ; j++ )
            if (( pattern[ k ] + rank * ring) == j )
                if( ( j % (( int ) sqrt( j )) ) != 0  )
                    prime = j;
    }
```

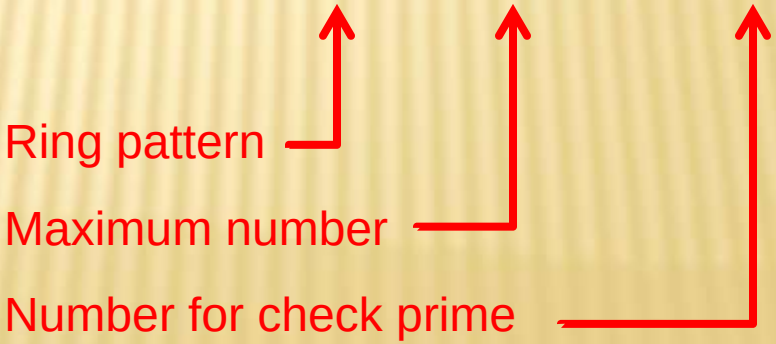
MPI COMPILE AND EXEXCUTION

MPI compile command :

 # mpicc –lm wheel_sieve.c

MPI execution :

 # mpiexec –n 4 ./a.out 30 2000 1561



REFERENCES

- http://en.wikipedia.org/wiki/Wheel_factorization
- A FULLY DISTRIBUTED PRIME NUMBERS GENERATION USING THE WHEEL SIEVE , Gabriel Paillard