

Part 2

'''

This script makes use of multiprocessing.pool in Python. Using ThreadPool, many threads can split up the process of determining the highest prime number in an array of N numbers.

In order to execute each array in parallel, it first creates random arrays of N integers for various values of N.

It simultaneously determines the greatest prime number in each chunk by using a ThreadPool with the number of threads indicated by the variable T.

The script then uses the variables T, N_vl, ch_size, a_s, m_ps, h_p, t_t, t1, and t2 to display the greatest prime number obtained and the total runtime for each value of N.

By using this method, the script makes effective use of the CPU resources that are available, enhancing performance through multithreading and parallelism.

'''

```
import numpy as np
```

```
import time
```

```
from multiprocessing.pool import ThreadPool as Pool
```

```
def is_prm(n):
```

```
    if n <= 1:
```

```
        return False
```

```
    if n == 2:
```

```
        return True
```

```
    if n % 2 == 0: # n is divisible or not
```

```
        return False
```

```
    for i in range(3, int(n**0.5) + 1, 2): # This runs for sqrt(n) times.
```

```
        if n % i == 0:
```

```
            return False
```

```
    return True
```

```
def find_h_prm(a):
```

```
    m_p = None
```

```
    for n in a:
```

```
        if is_prm(n):
```

```
            if m_p is None or n > m_p:
```

```
                m_p = n
```

```
    return m_p
```

```
T = 4 # Number of threads = 4
```

```

N_vl = [1000, 5000, 10000, 50000, 121000]

# this runs for various N values
for N in N_vl:
    print(f"\nRunning for N = {N}")
    ar_y = np.random.randint(1, 10000, N)
    ch_size = len(ar_y) // T
    a_s = [ar_y[i:i+ch_size] for i in range(0, len(ar_y), ch_size)]
    t1 = time.time()
    wr = Pool(T)    # inititalize the workers
    m_ps = wr.map(fnd_h_prm, a_s)
    wr.close()
    wr.join()
    h_p = max(m_ps)
    t2 = time.time()
    t_t = t2 - t1

    print("Highest Prime Number:", h_p)
    print("Total runtime:", t_t, "seconds")

```

```

Running for N = 1000
Highest Prime Number: 9949
Total runtime: 0.016861438751220703 seconds

```

```

Running for N = 5000
Highest Prime Number: 9973
Total runtime: 0.015014410018920898 seconds

```

```

Running for N = 10000
Highest Prime Number: 9973
Total runtime: 0.03719139099121094 seconds

```

```

Running for N = 50000
Highest Prime Number: 9973
Total runtime: 0.17368078231811523 seconds

```

```

Running for N = 121000
Highest Prime Number: 9973
Total runtime: 0.3822336196899414 seconds

```