Text

Description automatically generated

Zauważono, że wydajniej jest zliczać liczby, które nie są liczbami pierwszymi podczas pętli wykonującej wyszukiwanie liczb pierwszych, aniżeli zliczać liczby pierwsze w dodatkowej pętli, która musi przejść jeszcze raz po wynikowej tablicy.

Text

Description automatically generated

Zaimplementowano zliczanie liczb niepierwszych do podejścia sita równoległego domenowego. Na uwagę zasługuje linia kodu 301 w której od maxIndex odejmujemy 1. Takie postępowanie sprawia że wątki dostają rozłączne przedziały liczb. Bez tej korekty liczby graniczne ze zbiorów były przetwarzane przez dwa wątki, co prowadziło do do dublowanych zliczeń liczb niepierwszych w linii kodu 315.

Lokalność obliczeń liczb niepierwszych zapewniona przez deklarację zmiennej localNonPrimeCounter dla każdego wątku (linia kodu 293). Wyścigowi zapobiega atomowy dostęp wątków do zmiennej nonPrimeCounter która finalnie jest użyta do obliczenia ilości liczb pierwszych.

Próba implementacji zliczania liczb niepierwszych dla podejścia funkcyjnego kończy się fiaskiem. W podejściu funkcyjnym wszystkie wątki działają na wspólnym zbiorze liczb (arrayBoolean) dzieląc się dzielnikami (dividers), odwrotnie do podejścia domenowego. Dochodzi do wyścigu. Ponieważ w momencie, gdy wiele wątków przejdzie jednocześnie przez linię kodu 313 oznaczającą sprawdzenie, czy liczba jest zaznaczona jako pierwsza, każdy wątek będzie chciał ją oznaczyć jako niepierwszą oraz zwiększyć swój lokalny licznik liczb niepierwszych. Efektem jest uzyskiwanie różnych wyników.

Text

Description automatically generated

Rozwiązaniem jest implementacja standardowego licznika liczb pierwszych jako kolejną interację po tablicy wynikowej. Na uwagę zasługuje klauzula schedule(dynamic) w linii kodu 328. Bez niej wyniki były dużo słabsze.