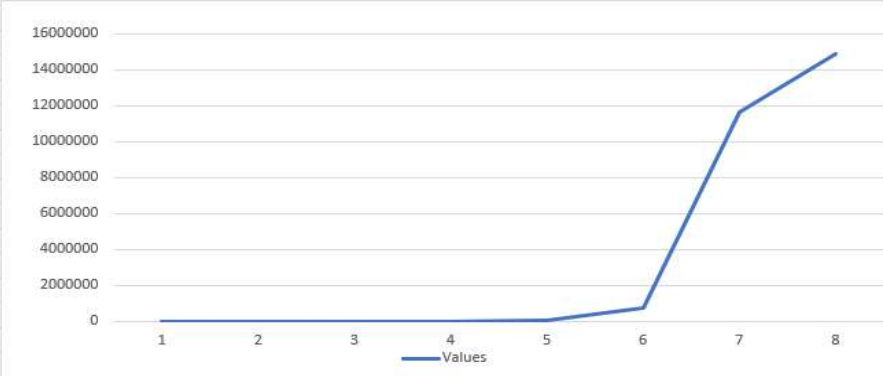
Piotr Waszak k37 Projekt 2

W tym projekcie, mamy za zadanie zaimplementowanie algorytmu sprawdzającego, czy liczba jest pierwsza. Podane jest osiem liczb, które należy sprawdzić algorytmem załączonym przez wykładowce, i dokonać instrumentacji. Następnie musimy zoptymalizować kod, tak aby proces wyszukiwania trwał krócej.

- Sprawdzanie liczb pierwszych bez instrumentacji
Prace z projektem, rozpocząłem od zaimplementowania kodu wykładowcy, kod dostępny na githubie pod nazwa program.cs .

Zmiany jakich dokonałem to stworzenie tablicy, z której pobierane są liczby do sprawdzenia. Dodatkowo dodałem na start pierwsza randomową liczbę, aby proces uruchamiania kodu, nie zabierał czasu z naszej właściwej liczby, abyśmy otrzymali rzetelny wynik.
Poniżej przedstawiony jest czas w jakim udało się sprawdzić wszystkie 8 liczb, oraz wykres

1	3
2	35
3	363
4	3536
5	34775
6	726540
7	11659376
8	14875495



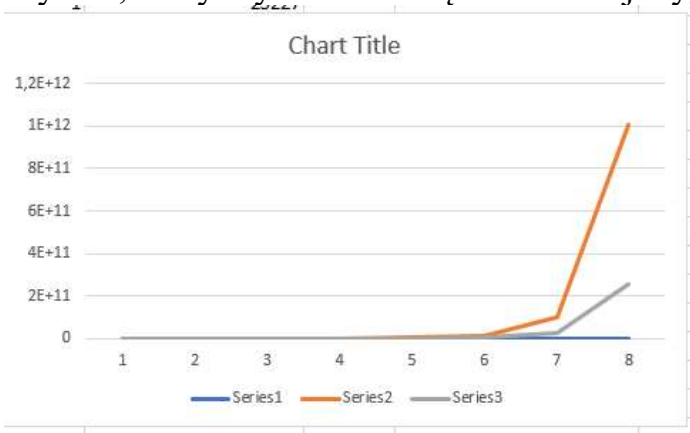
- Sprawdzanie liczb pierwszych z instrumentacja
na githubie pod nazwa : pierwszy_algorytm_instrumentacja

W algorytmie pod pętlą for, która za pomocą modulo dąży do 0, podpiąłem zinkremetnowany licznik (counter), który ma za zadanie sprawdzić ilość wykonania w tej pętli sprawdzenia. Counter został zaimplementowany jako public, aby był widoczny w naszym głównym kodzie, w którym go wyświetlamy.
Po wcześniejszym dlugim czasie oczekiwania, wykonałem teraz kod na 6 liczb, a następne 2 otrzymałem w wyniku proporcji:

D15 =(B15*D14)/B14

	A	B	C	D	E	F
1	1	25227				
2	2	252283				
3	3	2522849				
4	4	25228514				
5	5	252285151				
6	6	2522851532				
7	7	25228515322				
8	8	2,52285E+11				
9		10091406133		2522851532		
10		1,00914E+11		25228515322		
11						
12						
13						
14		1,00914E+11		25228515322		
15		1,00914E+12		252 285 153 227		
16						
17						

Wykres, który myślałem że będzie inaczej wyglądał.

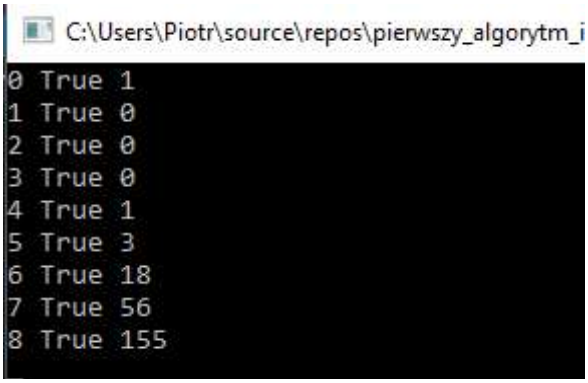


1	100 913	25 227
2	1 009 139	252 283
3	10 091 401	2 522 849
4	100 914 061	25 228 514
5	1 009 140 611	252 285 151
6	10 091 406 133	2 522 851 532
7	100 914 061 337	25 228 515 322
8	1 009 140 613 399	252 285 153 227

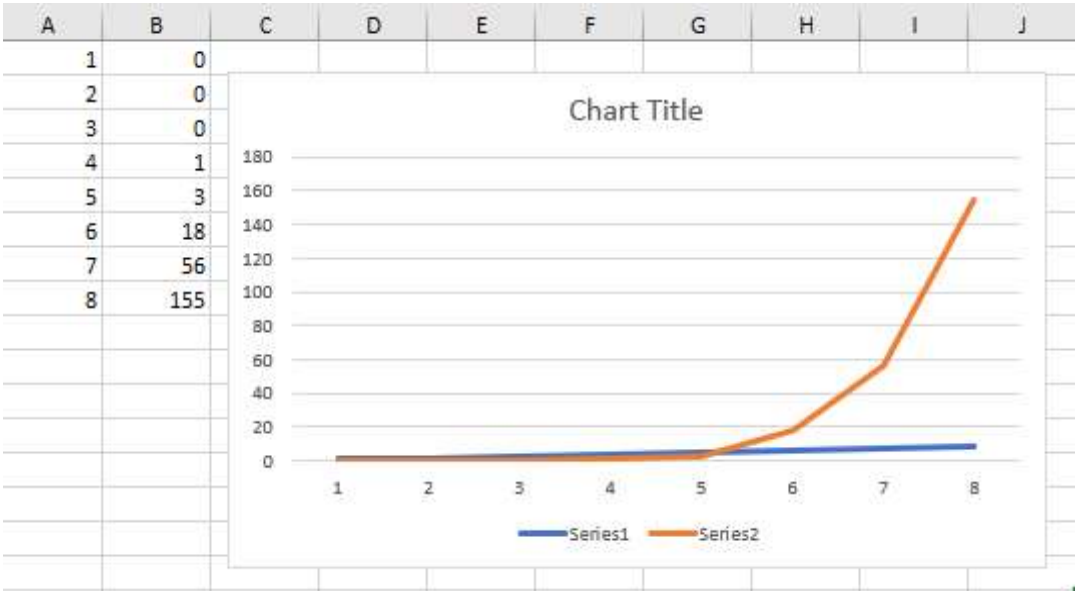
Series 2 to wykorzystane liczby pierwsze
Series 3 to wyniki instrumentacji

- Sprawdzanie liczb pierwszych po zmianie algorytmu bez instrumentacji
plik z kodem na githubie pod nazwa: drugi_algorytm_bez_instrumentacji

Instrukcja znacznie przyśpieszyła swoje działanie po zmianiu instrukcji w pętli for zamiast sprawdzania u <= liczba / 2. Na u <= pierwiastka liczby, zapisanego jako u *u <= liczba.

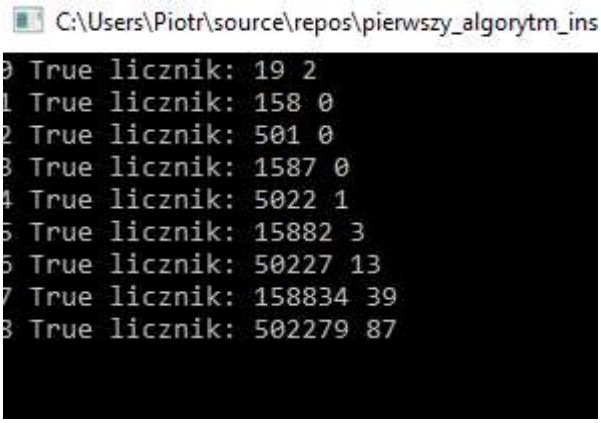


Wykres:

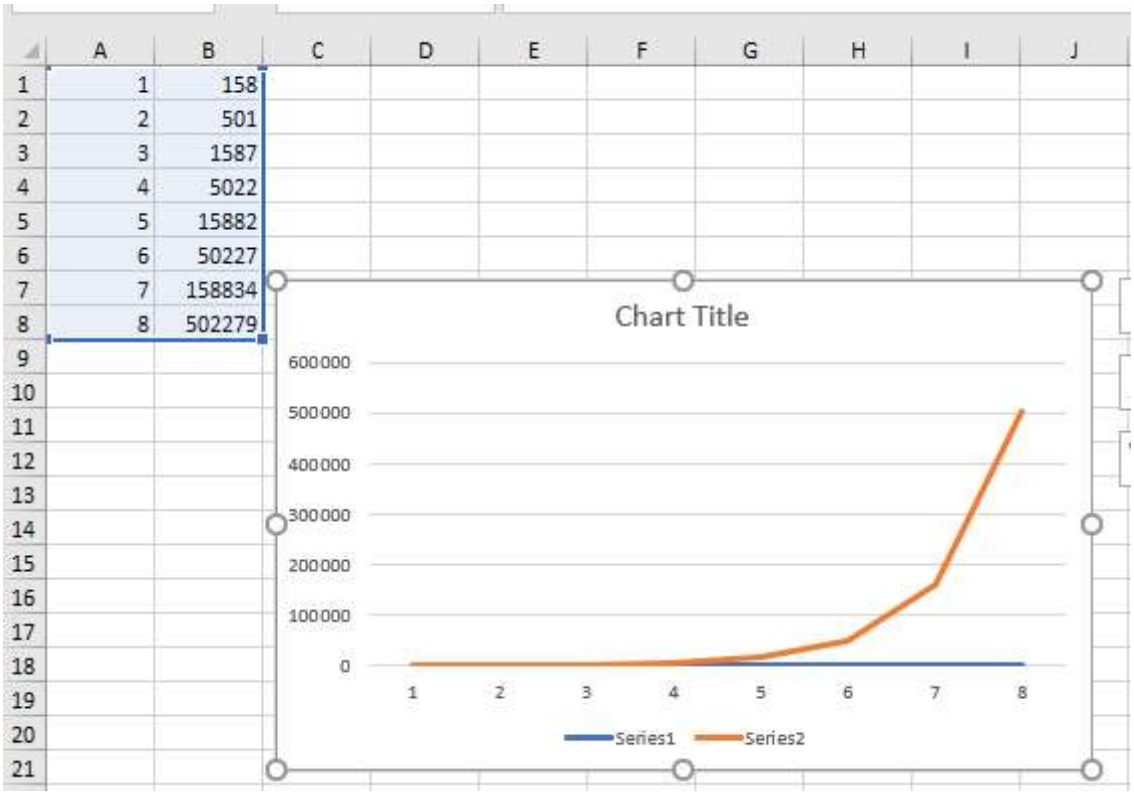


- Sprawdzanie liczb pierwszych po zmianie algorytmu z instrumentacja
plik z kodem na githubie pod nazwa: drugi_algorytm_instrumentacja

Przy poprzednim kodzie, liczniki instrumentacji zostały zakomentowane, w tym kodzie jedynie co zrobiłem to je odkomentowałem.



Wykres:



Podsumowując, druga metoda sprawdzania liczb pierwszych jest znacznie szybsza, zarówno jeśli chodzi o czas jak i ilość operacji.
W kodzie źródłowym, znajduję się komentarz odnośnie pierwszej liczby w tablicy, która nie będzie brana pod uwagę. Liczbe te po prostu zapomniałem zamienić w komentarzu.
Porównanie

1	100 913	25 227	158
2	1 009 139	252 283	501
3	10 091 401	2 522 849	1587
4	100 914 061	25 228 514	5022
5	1 009 140 611	252 285 151	15882
6	10 091 406 133	2 522 851 532	50227
7	100 914 061 337	25 228 515 322	158834
8	1 009 140 613 399	252 285 153 227	502279