Projekt 2

Badanie złożoności algorytmu sprawdzającego liczb pierwszych

Algorytm bez instrumentacji:

```
static bool isPrime(BigInteger Num)
{
   if (Num < 2) return false;
   else if (Num < 4) return true;
   else if (Num % 2 == 0) return false;
   else
      for (BigInteger u = 3; u < Num / 2; u += 2)
      {
         if (Num % u == 0) return false;
      }

// Poniższy kod jest optymalizacją powyższej pętli (przypadek przyzwoity)
      /* for (BigInteger u = 3; u * u <= Num / 2; u += 2)
      {
         if (Num % u == 0) return false;
      } */

   return true;
}
```

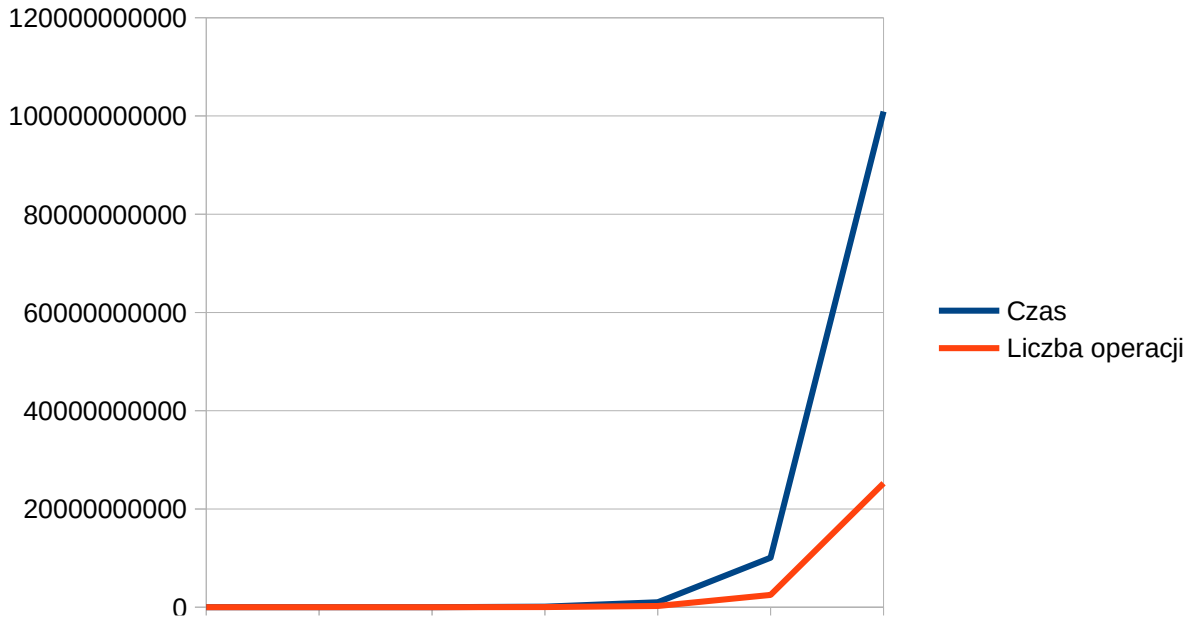
Algorytm z instrumentacją:

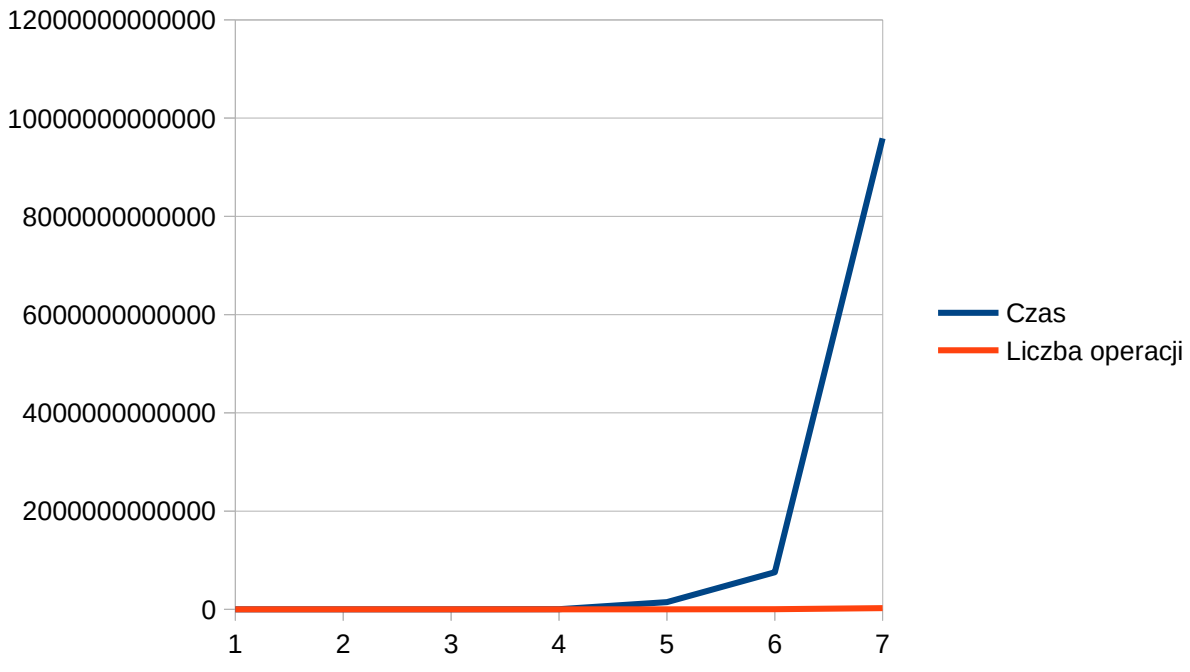
```
static int isPrimeInstr(BigInteger Num, int count)
{
   for (BigInteger u = 3; u < Num / 2; u += 2)
   {
      count++;
   }
   return count;
```

}

Przypadek podstawowy

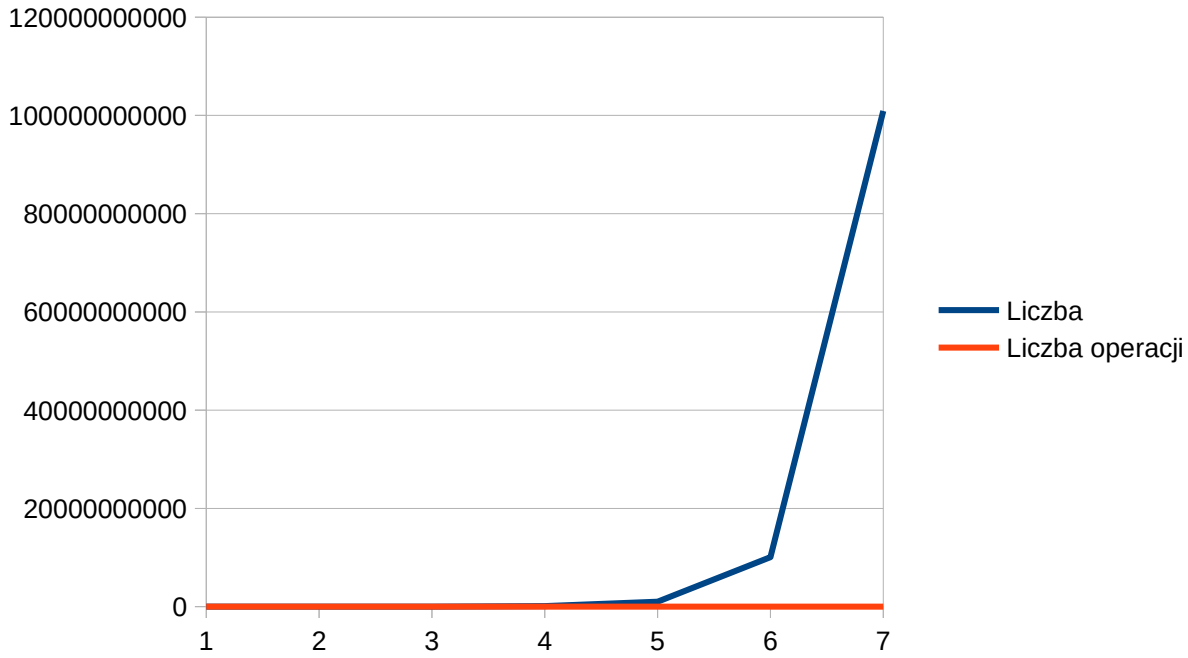
Liczba	Czas	Liczba operacji
100913	4668365	25228
1009139	41513012	252284
10091401	413148067	2522850
100914061	4141077451	25228515
1009140611	146970016504	252285152
10091406133	755800618234	2522851533
100914061337	9589924506878	25228515334

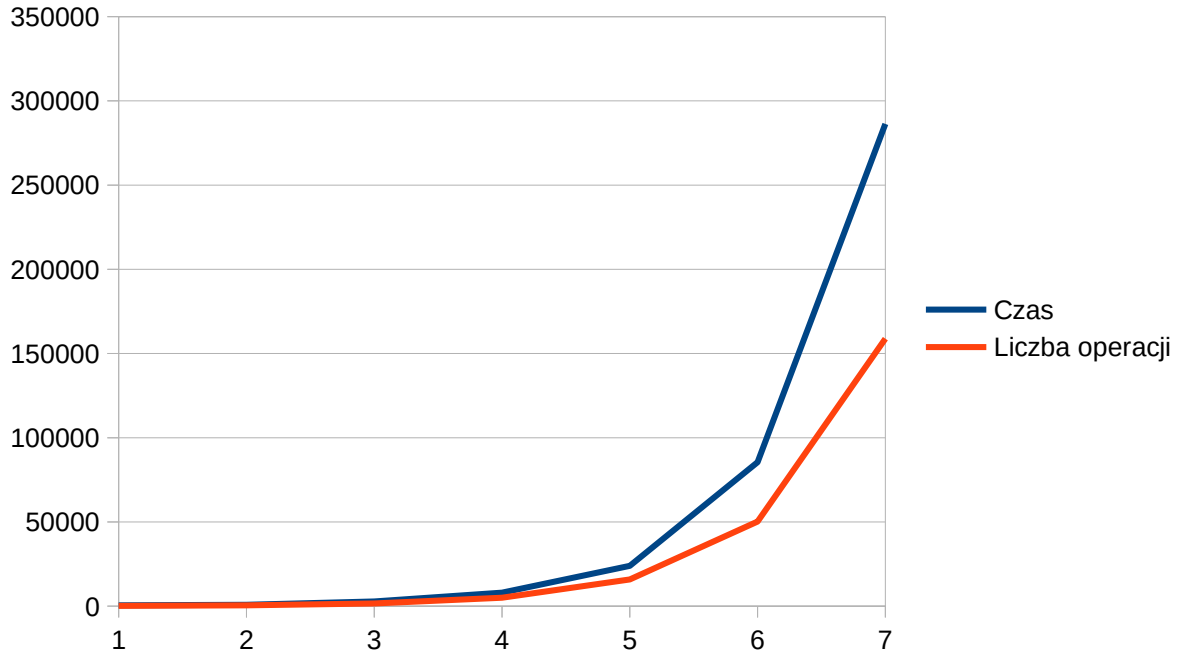




Przypadek przyzwoity

Liczba	Czas	Liczba operacji
100913	523	159
1009139	813	502
10091401	2756	1588
100914061	7997	5023
1009140611	23961	15883
10091406133	85466	50228
100914061337	286354	158835





Wnioski:

- Złożoność algorytmu jest pierwiastkowa.