Projekt 2

do samodzielnego wykonania

Dane jest poniższa implementacja algorytmu badania czy zadana liczba jest pierwsza:

```
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;
  return true;
}
```

Celem projektu jest zaproponowanie bardziej efektywnego algorytmu przy zachowaniu niezmienionego interfejsu podprogramu. Przeprowadzić analizę za pomocą instrumentacji i pomiarów czasu. Przyjąć, że operacją dominującą jest dzielenie modulo (%).

W sprawozdaniu przedstawić dla obu algorytmów:

- kod źródłowy przed instrumentacją
- kod źródłowy po instrumentacji
- zebrane wyniki w postaci tekstu i wykresów
- wnioski z analizy zebranych danych (ocena złożoności)

Badanie przeprowadzić dla następującego zbioru punktów pomiarowych (liczb pierwszych):
{ 100913, 1009139, 10091401, 100914061, 1009140611, 10091406133, 100914061337, 1009140613399 }

Algorytm wyszukiwania liczby pierwszej

Przykładowy

Funkcja bez instrumentacji

```
private static bool Pierwszabezinstrumentacji(BigInteger Num)// funckja bez instumentacji
{
    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; } }
    return true;
}
```

Funkcja z instrumentacją

```
private static bool Pierwszazinstrumentacja(BigInteger Num) // funkcja z instrumentacją
{
    Licznik = 1;
    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)
        {

            Licznik++; // liczy ilośc operacji
            if (Num % u == 0)

            {
                ;

                return false;

            }
        }
        return true;

    }
}
```

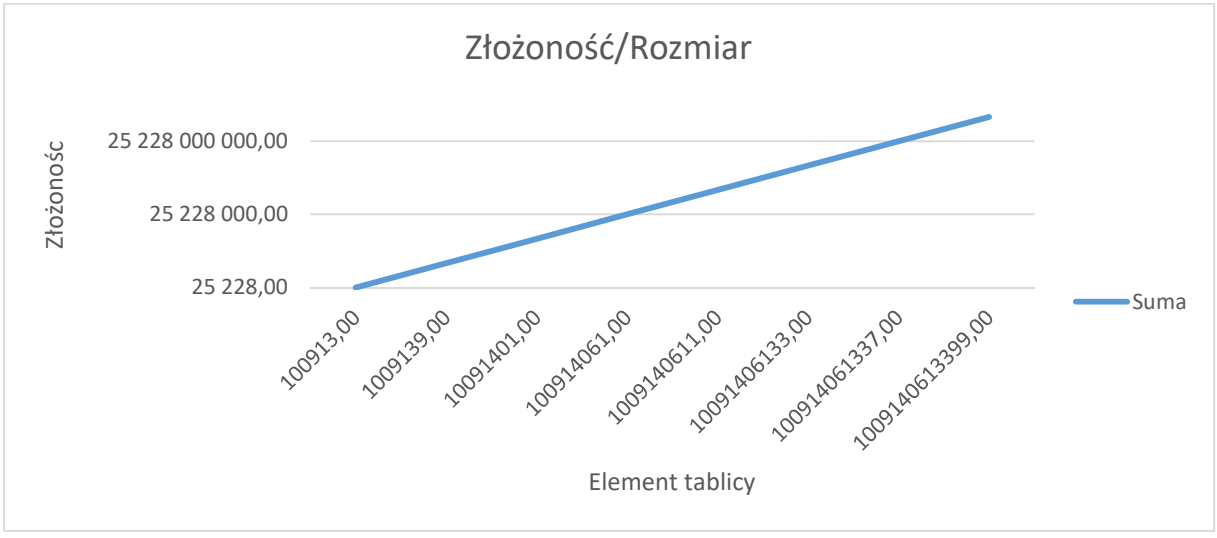
Funkcja główna

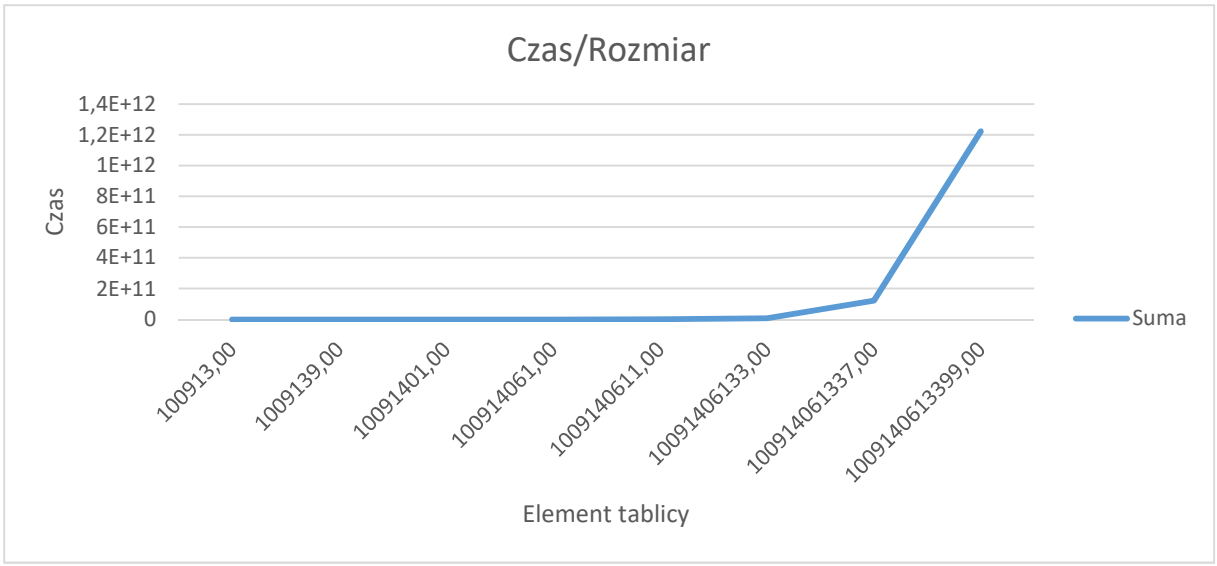
```
static void Main(string[] args)

{

    System.Console.WriteLine("NR;       Czas;       Złożoność");
    BigInteger[] Tablica = new BigInteger[] { 100913, 1009139, 10091401, 100914061, 1009140611, 10091406133, 100914061337, 1009140613399 };//tabela zawierająca wartości z góry podane


    for (int i = 0; i < Tablica.Length; i++)
    {
        var czas = new System.Diagnostics.Stopwatch();
        czas.Start();//czas start

        Pierwszabezinstrumentacji(Tablica[i]);//funkcja bez instrumentacji
        czas.Stop();// czas stop
        ulong czasroznica = (ulong)czas.ElapsedTicks; // obliczanie czasu w tikach procesora

        Wynik = Pierwszazinstrumentacja(Tablica[i]);

        System.Console.WriteLine((Tablica[i]) + ";" + czas.ElapsedTicks + ";" + Licznik);

    }


}
```

Wyniki algorytmu:

NR	Czas	Złożoność
100913,00	47500,00	25228,00
1009139,00	425161,00	252284,00
10091401,00	4572435,00	2522850,00
100914061,00	44542582,00	25228515,00
1009140611,00	424949055,00	252285152,00
10091406133,00	8202414093,00	2522851533,00
100914061337,00	122216774175,00	25228515334,00
1009140613399,00	1222167741785,00	252285153347,00

Wykresy:





Algorytm wyszukiwania liczby pierwszej

Przyzwoity

Funkcja bez instrumentacji:

```
private static bool Pierwszabezinstrumentacji(BigInteger Num)// funckja bez instumentacji
{
    if (Num < 2) { return false; }
    else if (Num < 4) { return true; }
    else if (Num % 2 == 0) { return false; }
    else
        for (BigInteger u = 3; u * u <= Num; u += 2)
        { if (Num % u == 0) { return false; } }
    return true;
}
```

Funkcja z instrumentacją:

```
private static bool Pierwszazinstrumentacja(BigInteger Num)// funckja z instumentacją
{
    Licznik = 1;
    if (Num < 2)
    { return false; }
    else if (Num < 4)
    { return true; }
    else if (Num % 2 == 0)
    { return false; }
    else
    {
        for (BigInteger u = 3; u * u <= Num; u += 2)
        {

            Licznik++;// liczy ilośc operacji
            if (Num % u == 0)

            {
                ;

                return false;

            }
        }
        return true;

    }
}
```

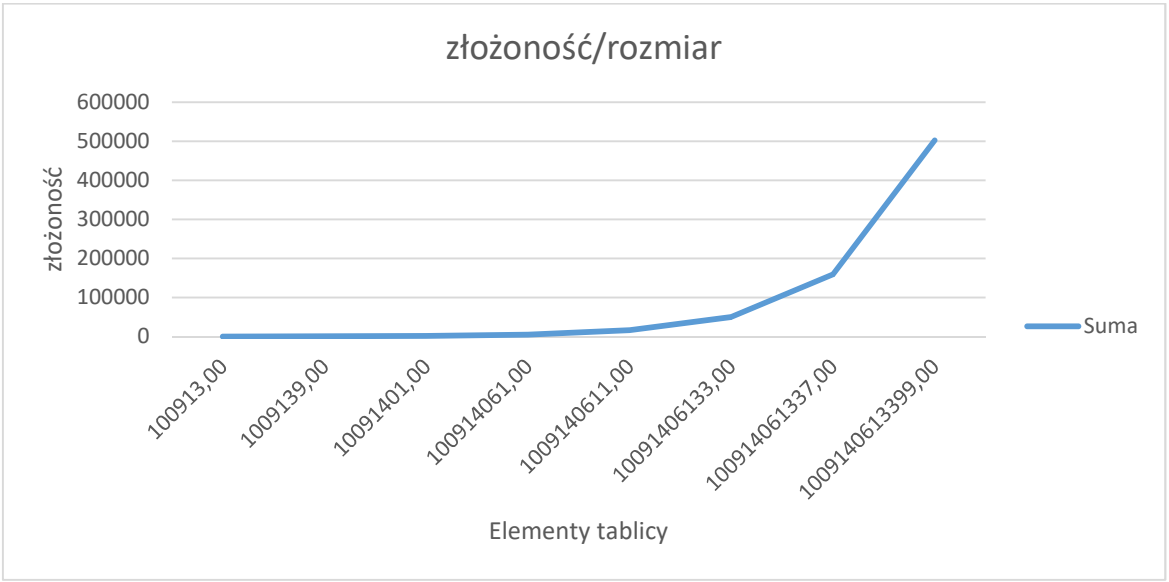
Funkcja główna:

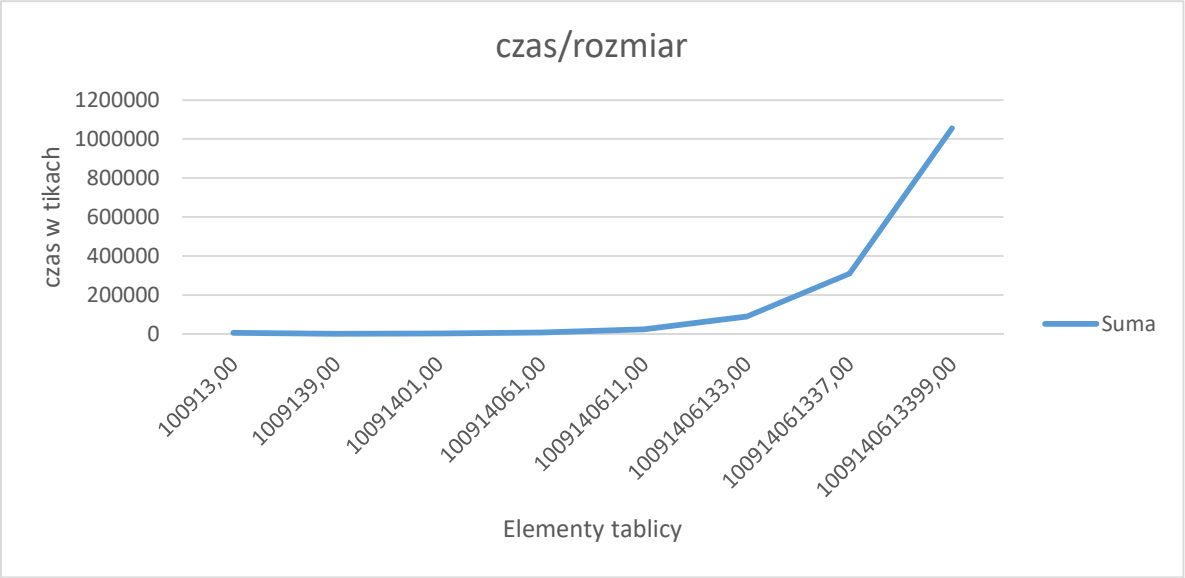
```
static void Main(string[] args)

{

    System.Console.WriteLine("NR;        Czas;        złożoność");
    BigInteger[] Tablica = new BigInteger[] { 100913, 1009139, 10091401, 100914061, 1009140611, 10091406133, 100914061337, 1009140613399 };//tabela zawierająca wartości z góry podane


    for (int i = 0; i < Tablica.Length; i++)
    {
        var czas = new System.Diagnostics.Stopwatch();
        czas.Start();// czas start

        Pierwszabezinstrumentacji(Tablica[i]);//funkcja bez instrumentacji
        czas.Stop();// czas stop
        ulong czasroznica = (ulong)czas.ElapsedTicks;// obliczanie czasu w tikach procesora

        Wynik = Pierwszazinstrumentacja(Tablica[i]);

        System.Console.WriteLine((Tablica[i]) + ";" + czas.ElapsedTicks + ";" + Licznik);

    }



    System.Console.ReadKey();
}
```

Wyniki algorytmu:

NR	Czas	Złożoność
100913,00	5802,00	159
1009139,00	950,00	502
10091401,00	2990,00	1588
100914061,00	8185,00	5023
1009140611,00	25046,00	15883
10091406133,00	89733,00	50228
100914061337,00	309804,00	158835
1009140613399,00	1055106,00	502280

Wykresy:





Komputer na którym wykonywano pomiary:

Wersja systemu Windows

Windows 10 Home

© 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

System

Procesor:	Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz 2.50 GHz
Zainstalowana pamięć (RAM):	8,00 GB (dostępne: 7,89 GB)
Typ systemu:	64-bitowy system operacyjny, procesor x64

Złożoność oraz czas 8 liczby algorytmu przykładowego obliczyłem za pomocą proporcji.

Czas liczyłem w tikach procesora.

Wydaje mi się że wysoki wynik czasu pierwszej liczby w algorytmie przyzwoitym jest wynikiem czasu oraz mocy obliczeniowej procesora na uruchomienie programu.

Wnioski:

1. Nie wielką zmianą w kodzie możemy sprawić aby nasz algorytm był o wiele bardziej wydajny pod względem czasu i liczby operacji.
2. Złożoność algorytmu przykładowego wynosi n/2 natomiast złożoność algorytmu przyzwoitego jest pierwiastkowa.