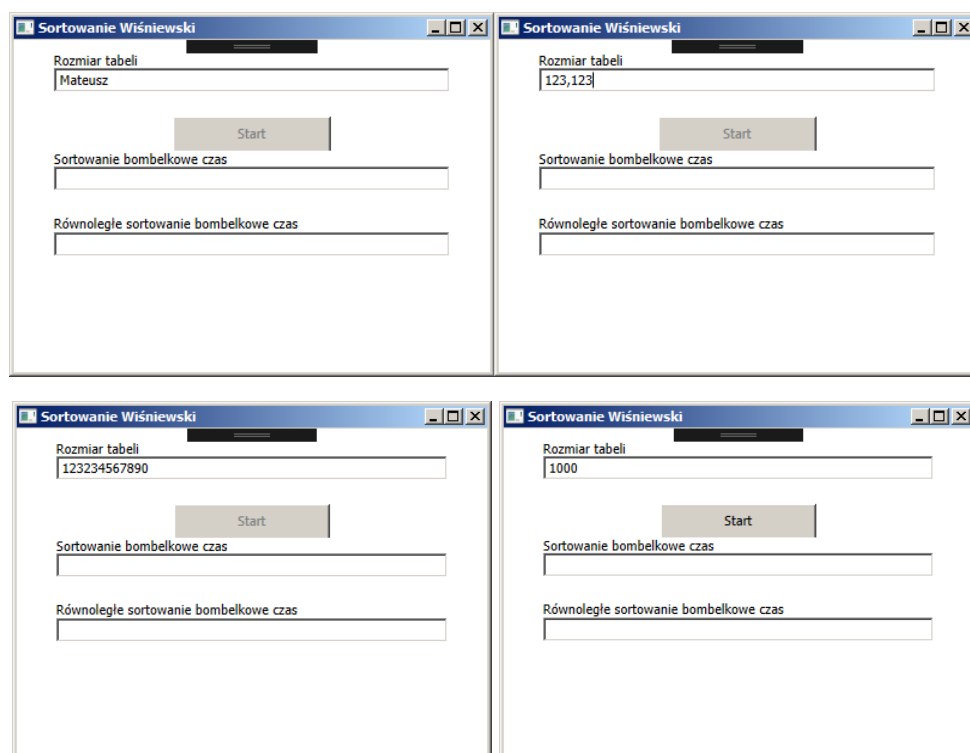


# Lab 3 Sortowanie równoległe

Mateusz Wiśniewski  
Politechnika Śląska

Zadanie polegało na zaprojektowaniu i prezentacji w formie graficznej wybranego algorytmu sortowania oraz porównanie czasu potrzebnego na posortowanie wygenerowanych danych dla standardowego sortowania oraz zrównoleglonego. Do realizacji zadania wybrałem sortowanie bąbelkowe.

Zadanie zostało wykonane za pomocą języka C# z wykorzystaniem technologii WPF oraz środowiska Microsoft Visual Studio 2017. Program opiera się na trzech elementach typu TextBox, trzech elementach typu TextBlock służących jako opis TextBoxów oraz przycisku, który rozpoczyna działanie programu. Przycisk „Start” jest jednocześnie walidatorem wprowadzonych danych, jeżeli wprowadzimy tekst zamiast rozmiaru tabeli, bądź przekroczymy maksymalny rozmiar tabeli przycisk ma ustawiona wartość „IsEnabled” na „False”.



Po wprowadzeniu rozmiaru tablicy i naciśnięciu przycisku „Start” tablica zostaje zapełniona losowymi cyframi:

```
public int[] RandomTable()
{
    Random rand = new Random();
    for (int i = 0; i < Table.Length; i++)
    {
        Table[i] = rand.Next(10);
    }

    return Table;
}
```

Standardowe sortowanie opiera się na dwóch pętlach for, w których zachodzi algorytm sortowania bąbelkowego.

```
public void Sort()
{
    for (int i = 0; i < Table.Length; i++)
    {
        for (int j = 0; j < Table.Length - 1; j++)
        {
            if (Table[j] < Table[j + 1])
            {
                var tmp = Table[j];
                Table[j] = Table[j + 1];
                Table[j + 1] = tmp;
            }
        }
    }
}
```

Równoległe sortowanie wykorzystuje natomiast pętlę Parallel.For

```
public void ParallelSort()
{
    Parallel.For(0, Table.Length, i =>
    {
        for (int j = 0; j < Table.Length - 1; j++)
        {
            if (Table[j] < Table[j + 1])
            {
                var tmp = Table[j];
                Table[j] = Table[j + 1];
                Table[j + 1] = tmp;
            }
        }
    });
}
```

Testy oraz analiza wyników:

Rozmiar tablicy		
	Standardowe sortowanie	Równoległe sortowanie
10	00:00:00.0002832	00:00:00.0003579
1000	00:00:00.0063407	00:00:00.0063344
100000	00:01:02.3861597	00:00:29.2531876
10000000		
1000000000		

Tablice zawierające mało danych zostały szybciej posortowane za pomocą standardowego sortowania, przy tablicy zawierającej 1000 cyfr można zauważyć podobny czas sortowania, natomiast przy większej liczbie danych sortowanie równoległe przebiega o wiele szybciej, im większa tablica, tym różnice w czasie są o wiele bardziej widoczne na korzyść równoległego sortowania.