

Projekt 1

Zweryfikować przeprowadzoną analizę średniej i pesymistycznej złożoności wyszukiwania liniowego i binarnego za pomocą instrumentacji i pomiarów czasu.

I. Wyszukiwanie liniowe w posortowanej tablicy int wielkości 268435456 (2^{28}).

Metody badania wykorzystane do zbadania złożoności :

- pomiar czasu potrzebnego na odnalezienie liczby.
- zliczanie operacji dominujących. W tym przypadku operacji porównania (==) w funkcji IsPresent.

Wybrane punkty pomiarowe :

1	26843545
2	53687090
3	80530635
4	107374180
5	134217725
6	161061270
7	187904815
8	214748360
9	241591905
10	268435456

Koszt maksymalny wynosi : 268 435 457

Koszt średni : 134 217 728

$$(a_1 + a_n) / 2$$

$$(0 + 268435456) / 2$$

Kod źródłowy przed instrumentacją :

```
class Program
{
    static void Main(string[] args)
    {
        int[] Vector = new int[268435457];
        int[] Points = a();
        for (int i = 0; i < 268435457; i++)
        {
            Vector[i] = i;
        }
        for (int i = 0; i < Points.Length; i++)
        {
            IsPresent(Vector, Points[i]);
        }

    }

    static bool IsPresent(int[] Vector, int Number)
    {
        for (int i = 0; i < Vector.Length; i++)
        {
            if (Vector[i]==Number)
            {
                return true;
            }
        }

        return false;
    }

    static int[] a()
    {
        int[] a = new int[10];
        int b = 0;
        for (int i = 0; i < 10; i++)
        {
            b += 26843545;
            a[i] = b;
        }
        return a;
    }
}
```

Kod źródłowy po instrumentacji:

```
class Program
{
    static void Main(string[] args)
    {
        int[] Vector = new int[268435457];
        int[] Points = a();
        for (int i = 0; i < 268435457; i++)
        {
            Vector[i] = i;
        }
        for (int i = 0; i < Points.Length; i++)
        {
            long Counter;
            long StartTime = Stopwatch.GetTimestamp();

            IsPresent(Vector, Points[i], out Counter);

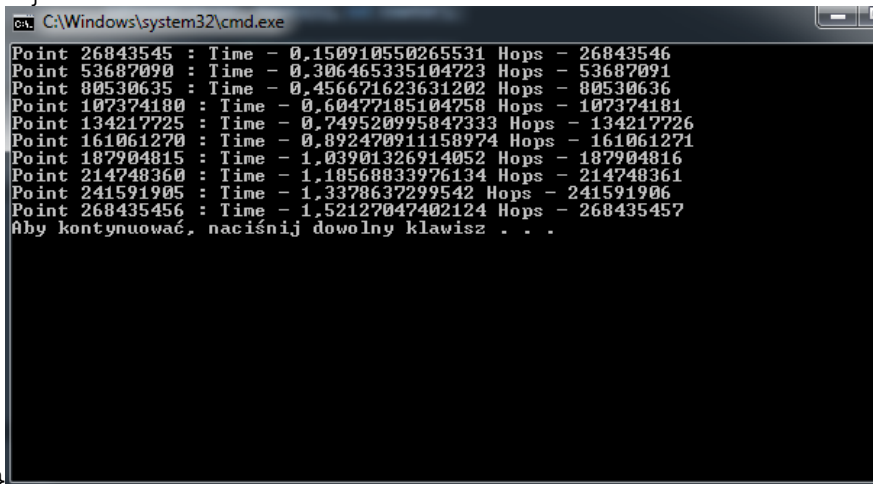
            long StopTime = Stopwatch.GetTimestamp();
            double ElapsedSeconds = (StopTime - StartTime) * (1.0 / Stopwatch.Frequency);

            Console.WriteLine($"Point {Points[i]} : Time - {ElapsedSeconds} Hops - {Counter}");
        }
    }

    static bool IsPresent(int[] Vector, int Number, out long a)
    {
        a = 0;
        for (long i = 0; i <= Vector.Length; i++)
        {
            if (Vector[i] == Number)
            {
                a = i+1 ;
                return true;
            }
        }

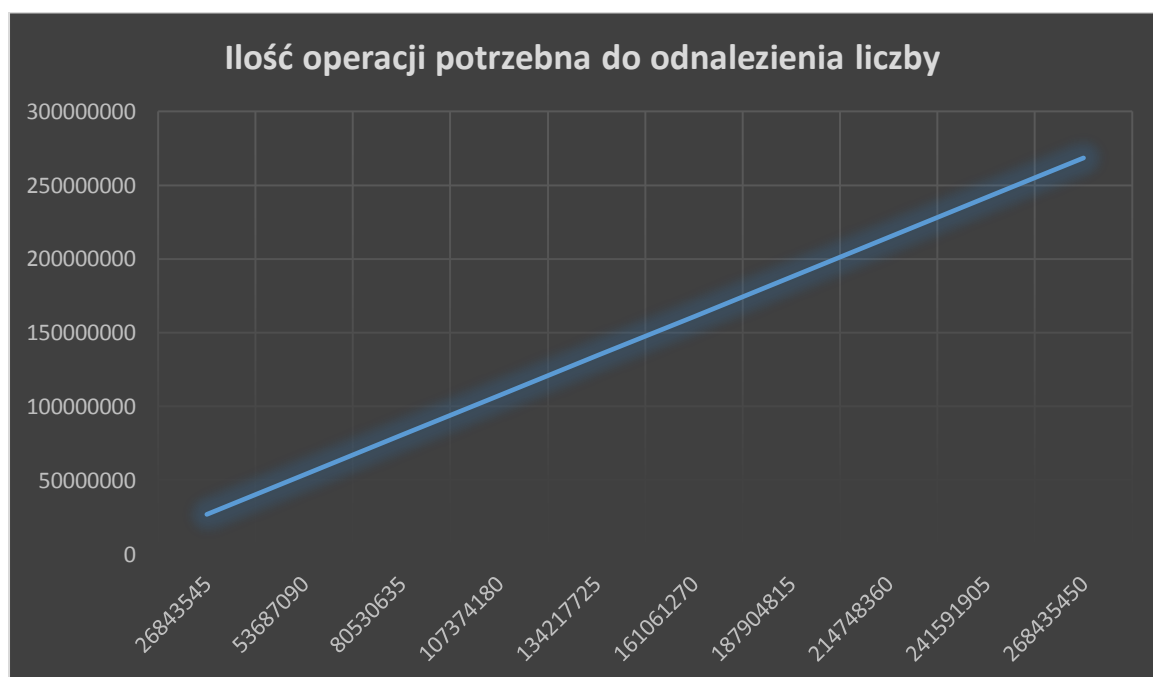
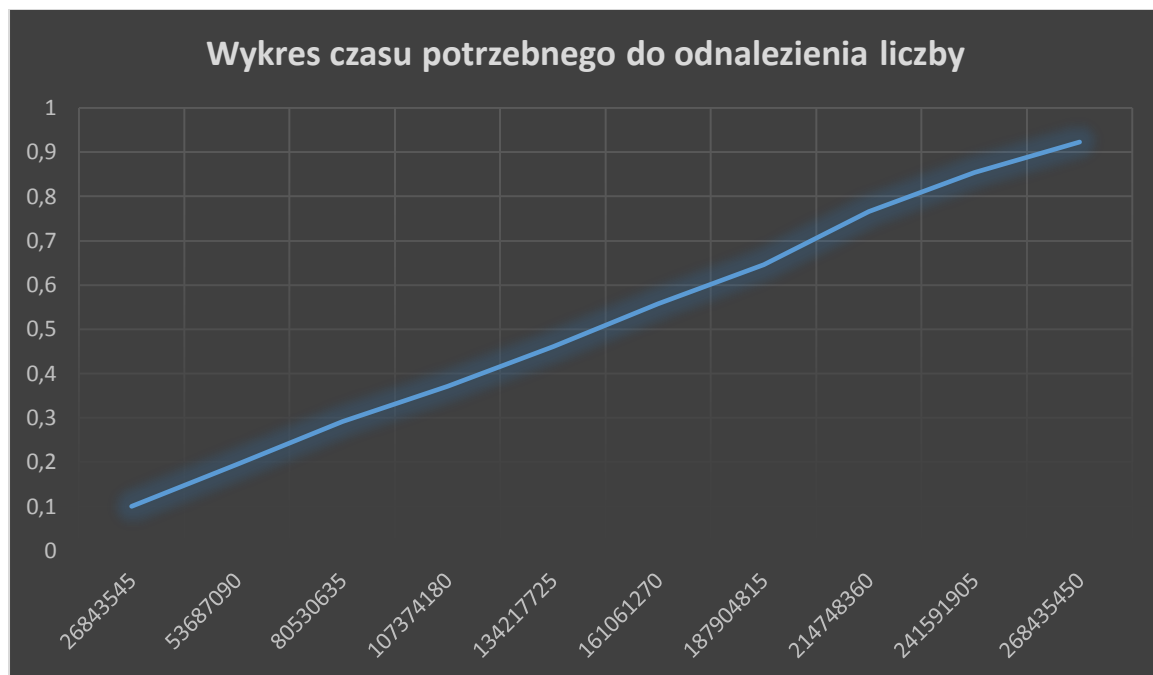
        return false;
    }

    static int[] a()
    {
        int[] a = new int[10];
        int b = 0;
        for (int i = 0; i < 9; i++)
        {
            b += 26843545;
            a[i] = b;
        }
        a[9] = 268435456;
        return a;
    }
}
```



```
C:\Windows\system32\cmd.exe
Point 26843545 : Time - 0.150910550265531 Hops - 26843546
Point 53687090 : Time - 0.306465335104723 Hops - 53687091
Point 80530635 : Time - 0.456671623631202 Hops - 80530636
Point 107374180 : Time - 0.60477185104758 Hops - 107374181
Point 134217725 : Time - 0.749520995847333 Hops - 134217726
Point 161061270 : Time - 0.892470911158974 Hops - 161061271
Point 187904815 : Time - 1.03901326914052 Hops - 187904816
Point 214748360 : Time - 1.18568833976134 Hops - 214748361
Point 241591905 : Time - 1.3378637299542 Hops - 241591906
Point 268435456 : Time - 1.52127047402124 Hops - 268435457
Aby kontynuować, naciśnij dowolny klawisz . . .
```

Procesor : Intel Core i5-2310



II. Wyszukiwanie binarnej posortowanej tablicy int wielkości $2^{28}-1$.

Metody badania wykorzystane do zbadania złożoności :

- pomiar czasu potrzebnego na odnalezienie liczby.
- zliczanie operacji kluczowych. W tym przypadku operacji porównania (==) w funkcji IsPresent.

Wybrane punkty pomiarowe :

1	134217727
2	67108863
3	33554431
4	16777215
5	8388607
6	4194303
7	2097151
8	1048575
9	524287
10	262143
11	131071
12	65535
13	32767
14	16383
15	8191
16	4095
17	2047
18	1023
19	511
20	255
21	127
22	63
23	31
24	15
25	7
26	3
27	1
28	0

Koszt maksymalny wynosi : 28

Koszt średni wynosi : 27
(Obliczenia w Excelu)

Liczba szukana	Liczba porównań	Liczba pozycji(mnożnik)	Wynik
134217727	1	1	1
67108863	2	2	4
33554431	3	4	12
16777215	4	8	32
8388607	5	16	80
4194303	6	32	192
2097151	7	64	448
1048575	8	128	1024
524287	9	256	2304
262143	10	512	5120
131071	11	1024	11264
65535	12	2048	24576
32767	13	4096	53248
16383	14	8192	114688
8191	15	16384	245760
4095	16	32768	524288
2047	17	65536	1114112
1023	18	131072	2359296
511	19	262144	4980736
255	20	524288	10485760
127	21	1048576	22020096
63	22	2097152	46137344
31	23	4194304	96468992
15	24	8388608	201326592
7	25	16777216	419430400
3	26	33554432	872415232
1	27	67108864	1811939328
0	28	134217728	3758096384
Suma			7247757313
Rozmiar tablicy			268435456
Średnia			27

Kod źródłowy przed instrumentacją :

```
class Program
{
    static void Main()
    {
        int[] Vector = new int[268435456];
        int[] Points = a();
        for (int i = 0; i < 268435456; i++)
        {
            Vector[i] = i;
        }
        for (int i = 0; i < Points.Length; i++)
        {
            IsPresent(Vector, Points[i]);
        }
    }

    static bool IsPresent(int[] Vector, int Number)
    {
        int Left = 0, Right = Vector.Length - 1, Middle;
        while (Left <= Right)
        {
            Middle = (Left + Right) / 2;
            if (Vector[Middle] == Number) return true;
            else if (Vector[Middle] > Number) Right = Middle - 1;
            else Left = Middle + 1;
        }
        return false;
    }

    static int[] a()
    {
        int[] a = new int[28];
        int[] b = new int[28];

        a[0] = 0;
        int Counter = 27;

        for (int i = 1; i < 28; i++)
        {
            a[i] = (a[i - 1] * 2) + 1;
        }
        for (int i = 0; i < a.Length; i++)
        {
            b[i] = a[Counter--];
        }
        return b;
    }
}
```

Kod źródłowy po instrumentacji:

```
class Program
{
    static void Main()
    {
        int[] Vector = new int[268435456];
        int[] Points = a();
        for (int i = 0; i < 268435456; i++)
        {
            Vector[i] = i;
        }
        for (int i = 0; i < Points.Length; i++)
        {
            int Counter;
            long StartTime = Stopwatch.GetTimestamp();

            IsPresent(Vector, Points[i], out Counter);

            long StopTime = Stopwatch.GetTimestamp();
            double ElapsedSeconds = (StopTime - StartTime) * (1.0 /
Stopwatch.Frequency);
            Console.WriteLine($"Point {Points[i]} : Time - {ElapsedSeconds*1000}
Hops - {Counter}");
            Console.WriteLine();
        }
    }

    static bool IsPresent(int[] Vector, int Number, out int Counter)
    {
        int Left = 0, Right = Vector.Length - 1, Middle;
        Counter = 0;
        while (Left <= Right)
        {
            Counter++;
            Middle = (Left + Right) / 2;
            if (Vector[Middle] == Number) return true;
            else if (Vector[Middle] > Number) Right = Middle - 1;
            else Left = Middle + 1;
        }
        return false;
    }

    static int[] a()
    {
        int[] a = new int[28];
        int[] b = new int[28];

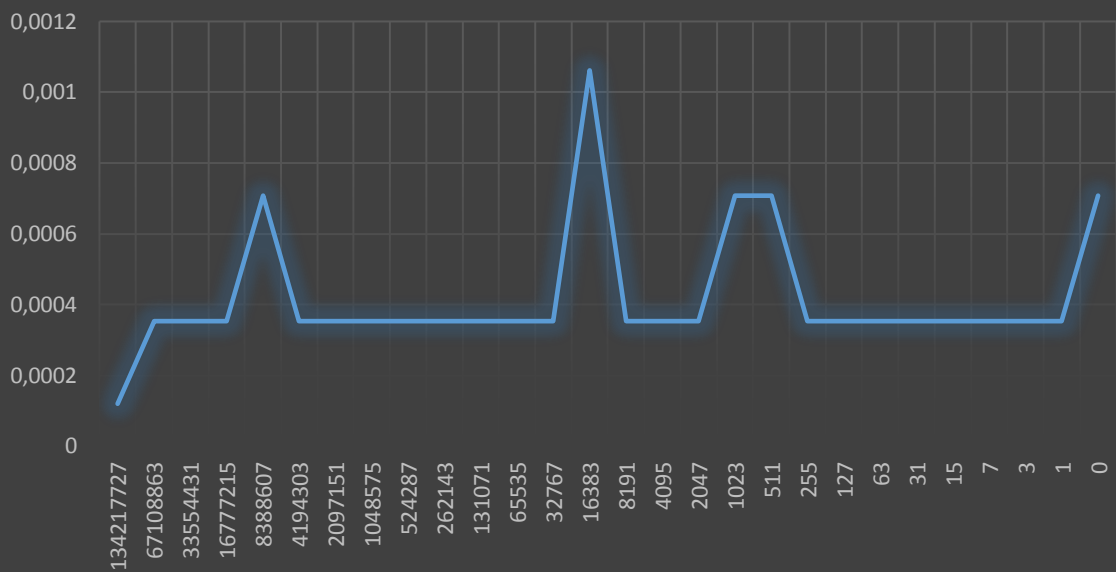
        a[0] = 0;
        int Counter = 27;

        for (int i = 1; i < 28; i++)
        {
            a[i] = (a[i - 1] * 2) + 1;
        }
        for (int i = 0; i < a.Length; i++)
        {
            b[i] = a[Counter--];
        }
        return b;
    }
}
```

```
ca: C:\Windows\system32\cmd.exe
Point 134217227 : Time - 0,00011995946004 Hops - 1
Point 67108863 : Time - 0,00035390037138305 Hops - 2
Point 33554431 : Time - 0,00035390037138305 Hops - 3
Point 16777215 : Time - 0,00035390037138305 Hops - 4
Point 8388607 : Time - 0,000707800742766099 Hops - 5
Point 4194303 : Time - 0,00035390037138305 Hops - 6
Point 2097151 : Time - 0,00035390037138305 Hops - 7
Point 1048575 : Time - 0,00035390037138305 Hops - 8
Point 524287 : Time - 0,00035390037138305 Hops - 9
Point 262143 : Time - 0,00035390037138305 Hops - 10
Point 131071 : Time - 0,00035390037138305 Hops - 11
Point 65535 : Time - 0,00035390037138305 Hops - 12
Point 32767 : Time - 0,00035390037138305 Hops - 13
Point 16383 : Time - 0,00106170111414915 Hops - 14
Point 8191 : Time - 0,00035390037138305 Hops - 15
Point 4095 : Time - 0,00035390037138305 Hops - 16
Point 2047 : Time - 0,00035390037138305 Hops - 17
Point 1023 : Time - 0,000707800742766099 Hops - 18
Point 511 : Time - 0,000707800742766099 Hops - 19
Point 255 : Time - 0,00035390037138305 Hops - 20
Point 127 : Time - 0,00035390037138305 Hops - 21
Point 63 : Time - 0,00035390037138305 Hops - 22
Point 31 : Time - 0,00035390037138305 Hops - 23
Point 15 : Time - 0,00035390037138305 Hops - 24
Point 7 : Time - 0,00035390037138305 Hops - 25
Point 3 : Time - 0,00035390037138305 Hops - 26
Point 1 : Time - 0,00035390037138305 Hops - 27
Point 0 : Time - 0,000707800742766099 Hops - 28
Aby kontynuować, naciśnij dowolny klawisz . . . _
```

Procesor : Intel Core i5-2310

Wykres czasu potrzebnego do odnalezienia liczby



Ilość operacji potrzebna do odnalezienia liczby

