# PROJEKT 1

**Zweryfikować przedstawioną na wykładzie ocenę średniej i pesymistycznej złożoności wyszukiwania liniowego i binarnego. Przeprowadzić analizę za pomocą instrumentacji i pomiarów czasu. W porównaniu wykorzystać tablice liczb całkowitych o rozmiarze rzędu 230 bajtów (228 elementów typu uint/int).**

## Parametry komputera użytego w badaniu

**Zainstalowany system operacyjny**
Microsoft Windows 10 Home PL (wersja 64-bitowa)
**Procesor**
Intel Core i5-8250U (4 rdzenie, od 1.6 GHz do 3.4 GHz, 6MB cache)
**Pamięć RAM**
8 GB (SO-DIMM DDR3, 1866 MHz)

## Algorytm wyszukiwania liniowego

Polega on na przeglądaniu kolejnych elementów danego zbioru danych w poszukiwaniu określonego elementu. Jeżeli przeglądany element posiada własności szukanego elementu, program zwraca jego indeks i kończy przeszukiwanie zbioru. Jeżeli nie, następuje kontynuacja przeszukiwania aż do ostatniego elementu w zbiorze.

W tym projekcie badany był przypadek średni (gdy szukany element występuje w zbiorze) oraz przypadek pesymistyczny (gdy szukany element znajduje się na ostatniej pozycji lub nie ma go w zbiorze).

## Implementacja metody wyszukiwania liniowego

```
static int Liniowe(int[] tabLin, int szukana)
        {

            for (int i = 0; i < tabLin.Length; i++)
            {


                if (tabLin[i] == szukana)
                {
                    return i;
                }
            }
            return -1;

        }
```

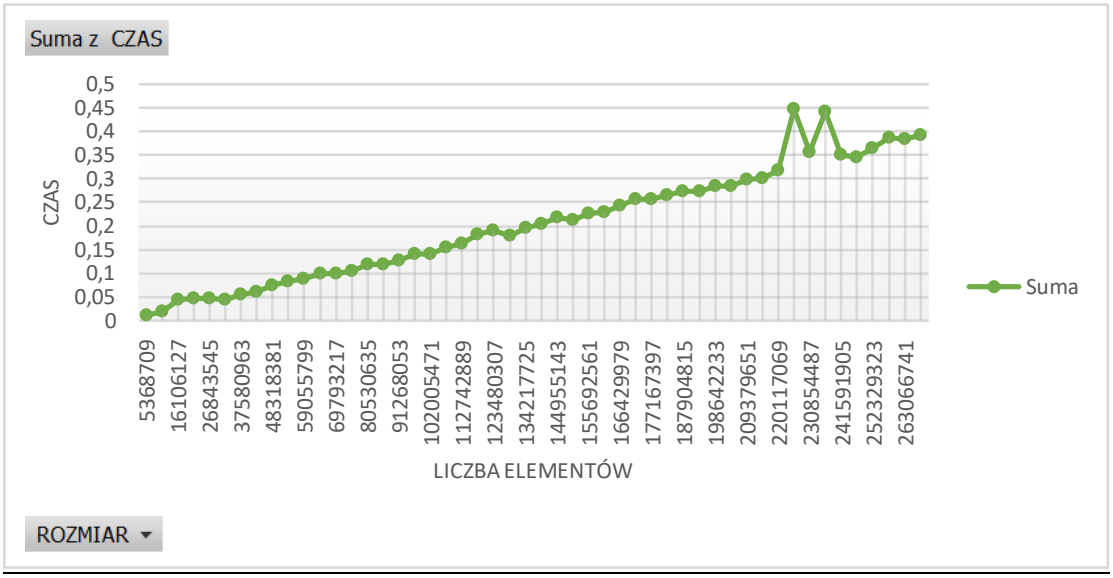
## Ocena przy wykorzystaniu instrumentacji – przypadek średni

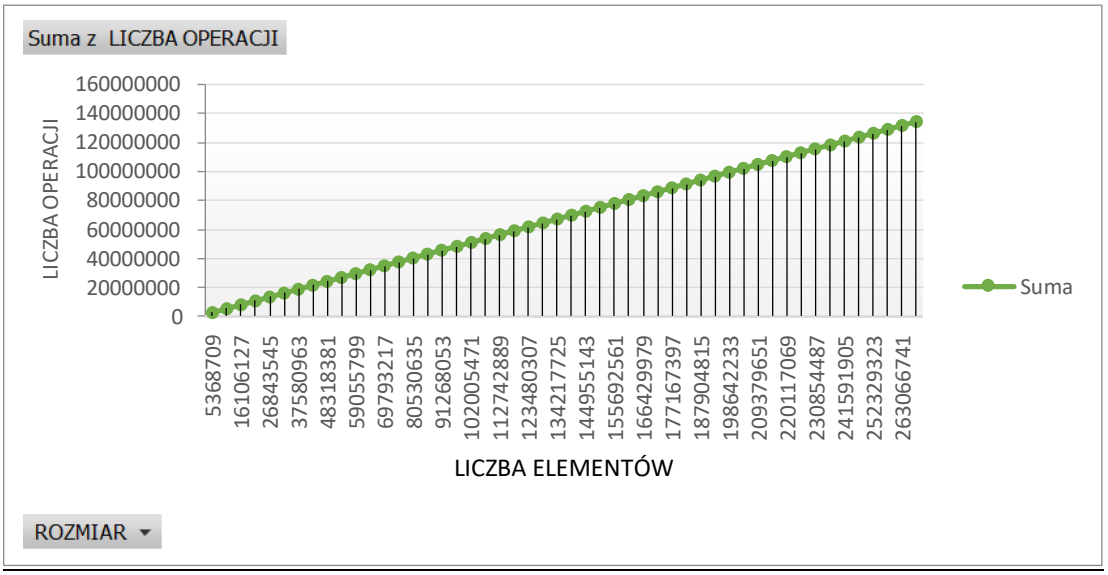
```
static int Instrumentacja(int[] tabLin, int szukana)
        {
            Suma = 0;
            Licznik = 0;
            for (int i = 0; i < tabLin.Length; i++)
            {
                ++Licznik;
                Suma += tabLin[i];
                Wynik = Suma / Licznik;

                if (tabLin[i] == szukana)
                {
                    return i;
                }
            }
            return -1;

        }
```

Obliczanie czasu przed instrumentacją

```
long StartingTime = Stopwatch.GetTimestamp();
                Liniowe(dane,wyszukiwanaSrednia);
                long EndingTime = Stopwatch.GetTimestamp();
                long ElapsedTime = EndingTime - StartingTime;
                double ElapsedSeconds = ElapsedTime * (1.0 / Stopwatch.Frequency);
```

WYKRESY



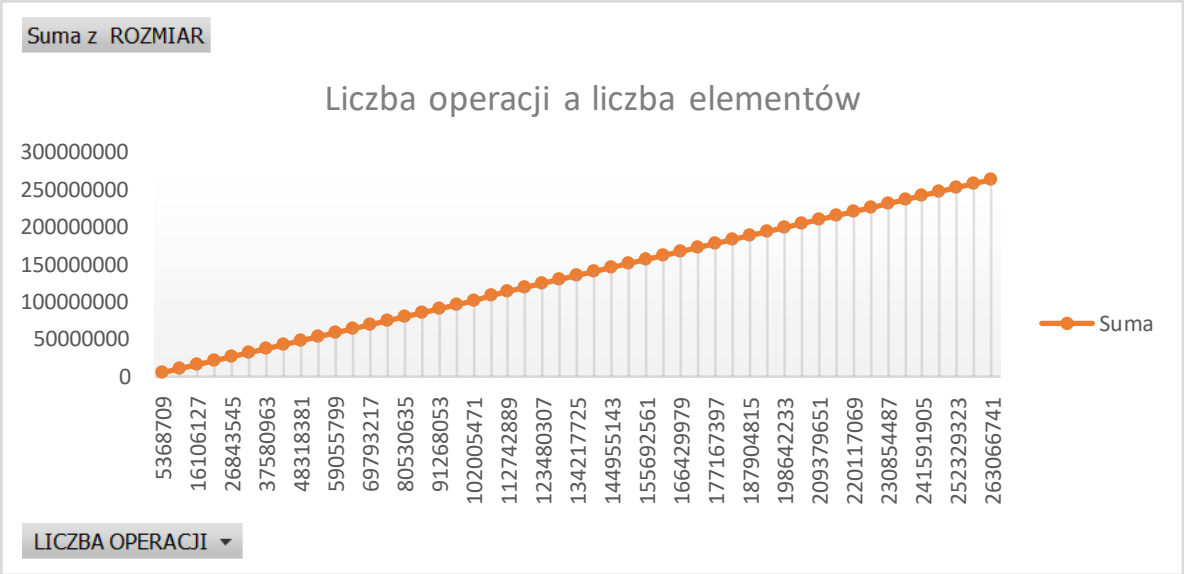
WYNIKI

| ELEMENT | POZYCJA | LICZBA OPERACJI | CZAS | ROZMIAR | ZŁOŻONOŚĆ |
|---|---|---|---|---|---|
| 2684354 | 2684353 | 2684354 | 0,0097095 | 5368709 | 1342177 |
| 5368709 | 5368708 | 5368709 | 0,018629 | 10737418 | 2684355 |
| 8053063 | 8053062 | 8053063 | 0,0443325 | 16106127 | 4026532 |
| 10737418 | 10737417 | 10737418 | 0,0454688 | 21474836 | 5368709 |
| 13421772 | 13421771 | 13421772 | 0,0471868 | 26843545 | 6710886 |
| 16106127 | 16106126 | 16106127 | 0,0444662 | 32212254 | 8053064 |
| 18790481 | 18790480 | 18790481 | 0,0562429 | 37580963 | 9395241 |
| 21474836 | 21474835 | 21474836 | 0,0591504 | 42949672 | 10737418 |
| 24159190 | 24159189 | 24159190 | 0,0742072 | 48318381 | 12079595 |
| 26843545 | 26843544 | 26843545 | 0,0832844 | 53687090 | 13421773 |
| 29527899 | 29527898 | 29527899 | 0,0889693 | 59055799 | 14763950 |
| 32212254 | 32212253 | 32212254 | 0,0985399 | 64424508 | 16106127 |
| 34896608 | 34896607 | 34896608 | 0,100154 | 69793217 | 17448304 |
| 37580963 | 37580962 | 37580963 | 0,1039929 | 75161926 | 18790482 |
| 40265317 | 40265316 | 40265317 | 0,118176 | 80530635 | 20132659 |
| 42949672 | 42949671 | 42949672 | 0,1196648 | 85899344 | 21474836 |
| 45634026 | 45634025 | 45634026 | 0,1269114 | 91268053 | 22817013 |
| 48318381 | 48318380 | 48318381 | 0,140572 | 96636762 | 24159191 |
| 51002735 | 51002734 | 51002735 | 0,1415205 | 102005471 | 25501368 |
| 53687090 | 53687089 | 53687090 | 0,1546842 | 107374180 | 26843545 |
| 56371444 | 56371443 | 56371444 | 0,1618326 | 112742889 | 28185722 |
| 59055799 | 59055798 | 59055799 | 0,1809094 | 118111598 | 29527900 |
| 61740153 | 61740152 | 61740153 | 0,1899787 | 123480307 | 30870077 |
| 64424508 | 64424507 | 64424508 | 0,1796629 | 128849016 | 32212254 |
| 67108862 | 67108861 | 67108862 | 0,1971137 | 134217725 | 33554431 |
| 69793217 | 69793216 | 69793217 | 0,2055595 | 139586434 | 34896609 |
| 72477571 | 72477570 | 72477571 | 0,2171807 | 144955143 | 36238786 |
| 75161926 | 75161925 | 75161926 | 0,213269 | 150323852 | 37580963 |
| 77846280 | 77846279 | 77846280 | 0,2275912 | 155692561 | 38923140 |
| 80530635 | 80530634 | 80530635 | 0,2285126 | 161061270 | 40265318 |
| 83214989 | 83214988 | 83214989 | 0,2438347 | 166429979 | 41607495 |
| 85899344 | 85899343 | 85899344 | 0,2558134 | 171798688 | 42949672 |
| 88583698 | 88583697 | 88583698 | 0,2565348 | 177167397 | 44291849 |
| 91268053 | 91268052 | 91268053 | 0,2641152 | 182536106 | 45634027 |
| 93952407 | 93952406 | 93952407 | 0,2738563 | 187904815 | 46976204 |
| 96636762 | 96636761 | 96636762 | 0,2729421 | 193273524 | 48318381 |
| 99321116 | 99321115 | 99321116 | 0,2838322 | 198642233 | 49660558 |
| 102005471 | 102005470 | 102005471 | 0,2848956 | 204010942 | 51002736 |
| 104689825 | 104689824 | 104689825 | 0,2981529 | 209379651 | 52344913 |
| 107374180 | 107374179 | 107374180 | 0,3000581 | 214748360 | 53687090 |
| 110058534 | 110058533 | 110058534 | 0,3187231 | 220117069 | 55029267 |
| 112742889 | 112742888 | 112742889 | 0,4464717 | 225485778 | 56371445 |
| 115427243 | 115427242 | 115427243 | 0,3574716 | 230854487 | 57713622 |
| 118111598 | 118111597 | 118111598 | 0,4421675 | 236223196 | 59055799 |
| 120795952 | 120795951 | 120795952 | 0,3513797 | 241591905 | 60397976 |
| 123480307 | 123480306 | 123480307 | 0,3450096 | 246960614 | 61740154 |
| 126164661 | 126164660 | 126164661 | 0,3646509 | 252329323 | 63082331 |
| 128849016 | 128849015 | 128849016 | 0,3880838 | 257698032 | 64424508 |
| 131533370 | 131533369 | 131533370 | 0,3827652 | 263066741 | 65766685 |
| 134217725 | 134217724 | 134217725 | 0,3911745 | 268435450 | 67108863 |

Ocena przy wykorzystaniu instrumentacji – przypadek pesymistyczny

```
static int Instrumentacja(int[] tabLin, int szukana)
        {
            Suma = 0;
            Licznik = 0;
            for (int i = 0; i < tabLin.Length; i++)
            {
                ++Licznik;

                if (tabLin[i] == szukana)
                {
                    return i;
                }
            }
            return -1;

        }
```

Wykresy




Wyniki

| ELEMENT | POZYCJA | LICZBA OPERACJI | CZAS | ROZMIAR | ZŁOŻONOŚĆ |
|---|---|---|---|---|---|
| 5368710 | -1 | 5368709 | 0,0156732 | 5368709 | 0 |
| 10737419 | -1 | 10737418 | 0,0391978 | 10737418 | 0 |
| 16106128 | -1 | 16106127 | 0,0679319 | 16106127 | 0 |
| 21474837 | -1 | 21474836 | 0,095584 | 21474836 | 0 |
| 26843546 | -1 | 26843545 | 0,1029285 | 26843545 | 0 |
| 32212255 | -1 | 32212254 | 0,1314389 | 32212254 | 0 |
| 37580964 | -1 | 37580963 | 0,1391215 | 37580963 | 0 |
| 42949673 | -1 | 42949672 | 0,130104 | 42949672 | 0 |
| 48318382 | -1 | 48318381 | 0,1550518 | 48318381 | 0 |
| 53687091 | -1 | 53687090 | 0,1534922 | 53687090 | 0 |
| 59055800 | -1 | 59055799 | 0,1686525 | 59055799 | 0 |
| 64424509 | -1 | 64424508 | 0,1895546 | 64424508 | 0 |
| 69793218 | -1 | 69793217 | 0,2025291 | 69793217 | 0 |
| 75161927 | -1 | 75161926 | 0,2128213 | 75161926 | 0 |
| 80530636 | -1 | 80530635 | 0,2377769 | 80530635 | 0 |
| 85899345 | -1 | 85899344 | 0,2565558 | 85899344 | 0 |
| 91268054 | -1 | 91268053 | 0,258124 | 91268053 | 0 |
| 96636763 | -1 | 96636762 | 0,2778191 | 96636762 | 0 |
| 102005472 | -1 | 102005471 | 0,2925726 | 102005471 | 0 |
| 107374181 | -1 | 107374180 | 0,307321 | 107374180 | 0 |
| 112742890 | -1 | 112742889 | 0,3356828 | 112742889 | 0 |
| 118111599 | -1 | 118111598 | 0,4371875 | 118111598 | 0 |
| 123480308 | -1 | 123480307 | 0,3793921 | 123480307 | 0 |
| 128849017 | -1 | 128849016 | 0,5009655 | 128849016 | 0 |
| 134217726 | -1 | 134217725 | 0,3966884 | 134217725 | 0 |
| 139586435 | -1 | 139586434 | 0,4293865 | 139586434 | 0 |
| 144955144 | -1 | 144955143 | 0,4322083 | 144955143 | 0 |
| 150323853 | -1 | 150323852 | 0,6304055 | 150323852 | 0 |
| 155692562 | -1 | 155692561 | 0,4828437 | 155692561 | 0 |
| 161061271 | -1 | 161061270 | 0,4853098 | 161061270 | 0 |
| 166429980 | -1 | 166429979 | 0,6479768 | 166429979 | 0 |
| 171798689 | -1 | 171798688 | 0,4988728 | 171798688 | 0 |
| 177167398 | -1 | 177167397 | 0,5263588 | 177167397 | 0 |
| 182536107 | -1 | 182536106 | 0,5708547 | 182536106 | 0 |
| 187904816 | -1 | 187904815 | 0,5499879 | 187904815 | 0 |
| 193273525 | -1 | 193273524 | 0,6951494 | 193273524 | 0 |
| 198642234 | -1 | 198642233 | 0,7414862 | 198642233 | 0 |
| 204010943 | -1 | 204010942 | 0,6901894 | 204010942 | 0 |
| 209379652 | -1 | 209379651 | 0,6364143 | 209379651 | 0 |
| 214748361 | -1 | 214748360 | 0,6416049 | 214748360 | 0 |
| 220117070 | -1 | 220117069 | 0,6708156 | 220117069 | 0 |
| 225485779 | -1 | 225485778 | 0,649966 | 225485778 | 0 |
| 230854488 | -1 | 230854487 | 0,7919874 | 230854487 | 0 |
| 236223197 | -1 | 236223196 | 0,7025927 | 236223196 | 0 |
| 241591906 | -1 | 241591905 | 0,7109285 | 241591905 | 0 |
| 246960615 | -1 | 246960614 | 0,7505988 | 246960614 | 0 |
| 252329324 | -1 | 252329323 | 0,7556719 | 252329323 | 0 |
| 257698033 | -1 | 257698032 | 0,7350586 | 257698032 | 0 |
| 263066742 | -1 | 263066741 | 0,8345634 | 263066741 | 0 |

Wyszukiwanie binarne

Polega na wyszukaniu wybranego elementu w zbiorze posortowanych elementów. W pierwszej operacji wyznaczany jest element środkowy element zbioru. Następnie następuje porównanie czy szukany element jest mniejszy, większy czy równy środkowemu elementowi. Jeśli środkowy element jest równy szukanemu, zwracany jest indeks tego elementu. Jeżeli nie, program zawęża poszukiwania w zależności od tego czy środkowy element był większy lub mniejszy od szukanego. Algorytm powtarza się aż do znalezienia szukanej wartości lub do sprawdzenia całego zbioru.

Implementacja metody wyszukiwania binarnego

```
static int Binarne(int[] tabBin, int szukana)
        {
            int prawa = tabBin.Length - 1;
            int lewa = 0;
            int srodek;

            while (lewa <= prawa)
            {
                srodek = (lewa + prawa) / 2;
                if (tabBin[srodek] == szukana)
                {
                    return srodek;
                }
                else if (tabBin[srodek] < szukana)
                {
                    lewa = srodek + 1;
                }
                else
                {
                    prawa = srodek - 1;
                }
            }
            return -1;
        }
```

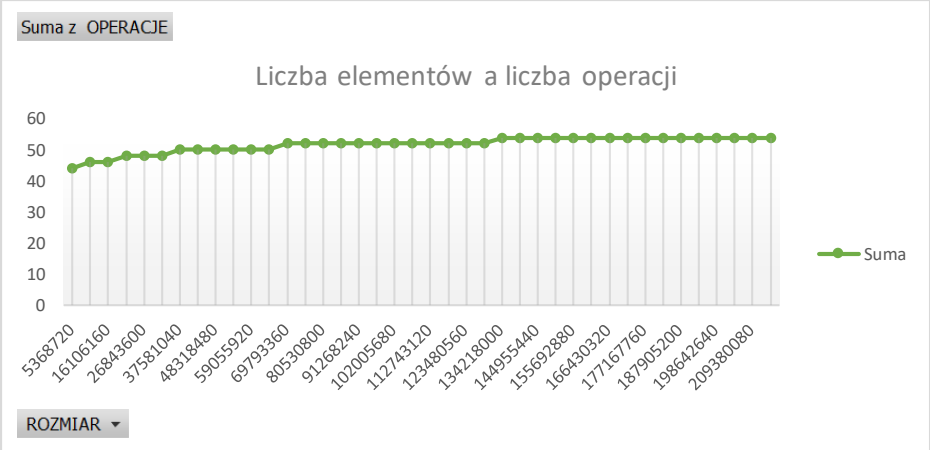
Ocena przy wykorzystaniu instrumentacji – przypadek średni

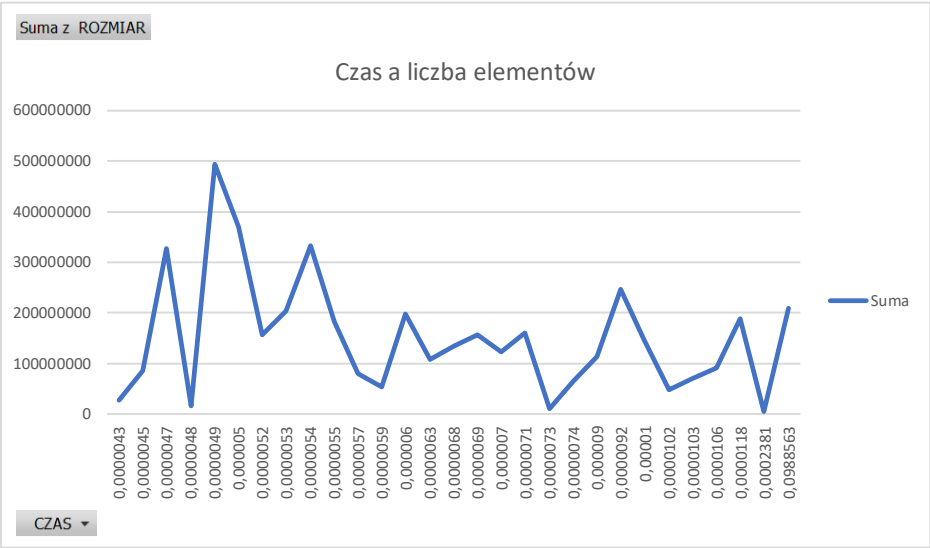
```
static int Instrumentacja(int[] tabBin, int szukana)
        {
            int prawa = tabBin.Length - 1;
            int lewa = 0;
            int srodek;
            Licznik = 0;
            dlugoscTab = 0;
            wynik = 0;
            while (lewa <= prawa)
            {
                ++Licznik;

                srodek = (lewa + prawa) / 2;
                wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);
                if (tabBin[srodek] == szukana)
                {
                    ++Licznik;
                    wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                    dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);
                    wynik = wynik / dlugoscTab;
                    return srodek;
                }
                else if (tabBin[srodek] < szukana)
                {
                    lewa = srodek + 1;
                    ++Licznik;
                    wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                    dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);
                }
                else
                {
                    prawa = srodek - 1;
                    ++Licznik;
                    wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                    dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);

                }
            }
            return -1;
        }
```

Obliczanie czasu przed instrumentacją

```
long StartingTime = Stopwatch.GetTimestamp();
                Binarne(dane, wyszukiwanaSrednia);
                long EndingTime = Stopwatch.GetTimestamp();
                long ElapsedTime = EndingTime - StartingTime;
                double ElapsedSeconds = ElapsedTime * (1.0 / Stopwatch.Frequency);
```

WYKRESY


Suma z OPERACJE
Liczba elementów a liczba operacji
60
50
40
30
20
10
0
5368720
16106160
26843600
37581040
48318480
59055920
69793360
80530800
91268240
102005680
112743120
123480560
134218000
144955440
155692880
166430320
177167760
187905200
198642640
209380080
Suma
ROZMIAR


Suma z ROZMIAR
Czas a liczba elementów
600000000
500000000
400000000
300000000
200000000
100000000
0
0,0000043
0,0000045
0,0000047
0,0000048
0,0000049
0,000005
0,0000052
0,0000053
0,0000054
0,0000055
0,0000057
0,0000059
0,000006
0,0000063
0,0000068
0,0000069
0,000007
0,0000071
0,0000073
0,0000074
0,000009
0,0000092
0,00001
0,0000102
0,0000103
0,0000106
0,0000118
0,0002381
0,0988563
Suma
CZAS

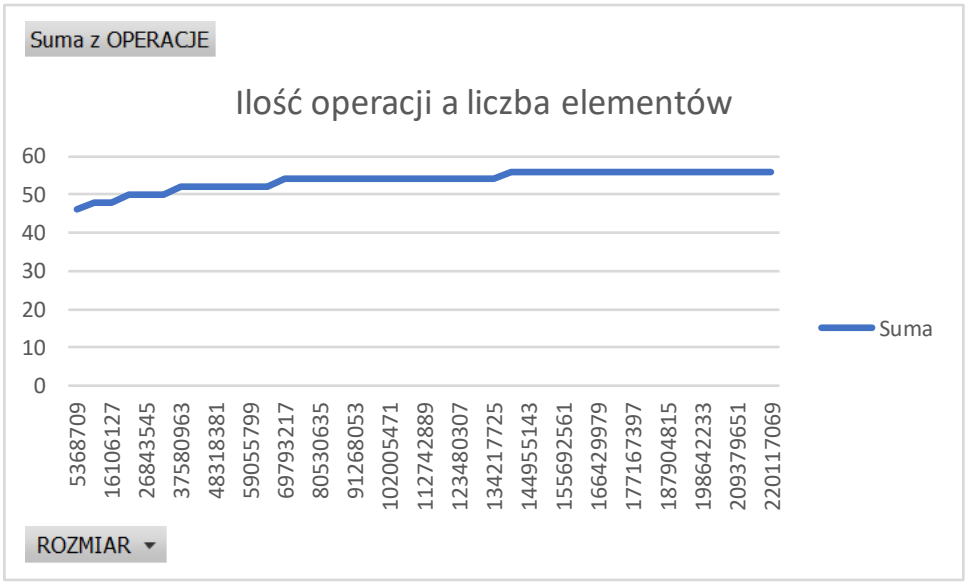
WYNIKI

| ELEMENT | POZYCJA | OPERACJE | CZAS | ROZMIAR | ZŁOŻONOŚĆ |
|---|---|---|---|---|---|
| 5368719 | 5368718 | 44 | 0,0002381 | 5368720 | 43 |
| 10737439 | 10737438 | 46 | 0,0000073 | 10737440 | 45 |
| 16106159 | 16106158 | 46 | 0,0000048 | 16106160 | 45 |
| 21474879 | 21474878 | 48 | 0,0000071 | 21474880 | 47 |
| 26843599 | 26843598 | 48 | 0,0000043 | 26843600 | 47 |
| 32212319 | 32212318 | 48 | 0,0000049 | 32212320 | 47 |
| 37581039 | 37581038 | 50 | 0,0000057 | 37581040 | 49 |
| 42949759 | 42949758 | 50 | 0,0000057 | 42949760 | 49 |
| 48318479 | 48318478 | 50 | 0,0000102 | 48318480 | 49 |
| 53687199 | 53687198 | 50 | 0,0000059 | 53687200 | 49 |
| 59055919 | 59055918 | 50 | 0,0000052 | 59055920 | 49 |
| 64424639 | 64424638 | 50 | 0,0000074 | 64424640 | 49 |
| 69793359 | 69793358 | 52 | 0,0000103 | 69793360 | 51 |
| 75162079 | 75162078 | 52 | 0,0000069 | 75162080 | 51 |
| 80530799 | 80530798 | 52 | 0,0000069 | 80530800 | 51 |
| 85899519 | 85899518 | 52 | 0,0000045 | 85899520 | 51 |
| 91268239 | 91268238 | 52 | 0,0000106 | 91268240 | 51 |
| 96636959 | 96636958 | 52 | 0,0000052 | 96636960 | 51 |
| 102005679 | 102005678 | 52 | 0,0000049 | 102005680 | 51 |
| 107374399 | 107374398 | 52 | 0,0000063 | 107374400 | 51 |
| 112743119 | 112743118 | 52 | 0,000009 | 112743120 | 51 |
| 118111839 | 118111838 | 52 | 0,0000092 | 118111840 | 51 |
| 123480559 | 123480558 | 52 | 0,000007 | 123480560 | 51 |
| 128849279 | 128849278 | 52 | 0,0000092 | 128849280 | 51 |
| 134217999 | 134217998 | 54 | 0,0000068 | 134218000 | 53 |
| 139586719 | 139586718 | 54 | 0,0000071 | 139586720 | 53 |
| 144955439 | 144955438 | 54 | 0,00001 | 144955440 | 53 |
| 150324159 | 150324158 | 54 | 0,0000047 | 150324160 | 53 |
| 155692879 | 155692878 | 54 | 0,000005 | 155692880 | 53 |
| 161061599 | 161061598 | 54 | 0,0000054 | 161061600 | 53 |
| 166430319 | 166430318 | 54 | 0,0000049 | 166430320 | 53 |
| 171799039 | 171799038 | 54 | 0,0000054 | 171799040 | 53 |
| 177167759 | 177167758 | 54 | 0,0000047 | 177167760 | 53 |
| 182536479 | 182536478 | 54 | 0,0000055 | 182536480 | 53 |
| 187905199 | 187905198 | 54 | 0,0000118 | 187905200 | 53 |
| 193273919 | 193273918 | 54 | 0,0000049 | 193273920 | 53 |
| 198642639 | 198642638 | 54 | 0,000006 | 198642640 | 53 |
| 204011359 | 204011358 | 54 | 0,0000053 | 204011360 | 53 |
| 209380079 | 209380078 | 54 | 0,0988563 | 209380080 | 53 |
| 214748799 | 214748798 | 54 | 0,000005 | 214748800 | 53 |

Ocena przy wykorzystaniu instrumentacji – przypadek pesymistyczny

```
static int Instrumentacja(int[] tabBin, int szukana)
        {
            int prawa = tabBin.Length - 1;
            int lewa = 0;
            int srodek;
            Licznik = 0;
            dlugoscTab = 0;
            wynik = 0;
            while (lewa <= prawa)
            {
                ++Licznik;

                srodek = (lewa + prawa) / 2;
                wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);
                if (tabBin[srodek] == szukana)
                {
                    ++Licznik;
                    wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                    dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);
                    wynik = wynik / dlugoscTab;
                    return srodek;
                }
                else if (tabBin[srodek] < szukana)
                {
                    lewa = srodek + 1;
                    ++Licznik;
                    wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                    dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);
                }
                else
                {
                    prawa = srodek - 1;
                    ++Licznik;
                    wynik += (ulong)Licznik * (ulong)Math.Pow(2, Licznik - 1);
                    dlugoscTab += (ulong)Math.Pow(2, Licznik - 1);

                }
            }
            return -1;
        }
```

WYKRESY




WYNIKI

| ELEMENT | POZYCJA | OPERACJE | CZAS | ROZMIAR |
|---|---|---|---|---|
| 5368710 | -1 | 46 | 0,000216 | 5368709 |
| 10737419 | -1 | 48 | 0,0000042 | 10737418 |
| 16106128 | -1 | 48 | 0,0000048 | 16106127 |
| 21474837 | -1 | 50 | 0,0000076 | 21474836 |
| 26843546 | -1 | 50 | 0,0000064 | 26843545 |
| 32212255 | -1 | 50 | 0,0000055 | 32212254 |
| 37580964 | -1 | 52 | 0,0000052 | 37580963 |
| 42949673 | -1 | 52 | 0,0000046 | 42949672 |
| 48318382 | -1 | 52 | 0,0000047 | 48318381 |
| 53687091 | -1 | 52 | 0,0000079 | 53687090 |
| 59055800 | -1 | 52 | 0,0000082 | 59055799 |
| 64424509 | -1 | 52 | 0,0000076 | 64424508 |
| 69793218 | -1 | 54 | 0,0000075 | 69793217 |
| 75161927 | -1 | 54 | 0,0000048 | 75161926 |
| 80530636 | -1 | 54 | 0,0351753 | 80530635 |
| 85899345 | -1 | 54 | 0,0000155 | 85899344 |
| 91268054 | -1 | 54 | 0,0000134 | 91268053 |
| 96636763 | -1 | 54 | 0,0000203 | 96636762 |
| 102005472 | -1 | 54 | 0,0000055 | 102005471 |
| 107374181 | -1 | 54 | 0,0598421 | 107374180 |
| 112742890 | -1 | 54 | 0,0000061 | 112742889 |
| 118111599 | -1 | 54 | 0,0000075 | 118111598 |
| 123480308 | -1 | 54 | 0,0000053 | 123480307 |
| 128849017 | -1 | 54 | 0,0000093 | 128849016 |
| 134217726 | -1 | 54 | 0,0617171 | 134217725 |
| 139586435 | -1 | 56 | 0,00001 | 139586434 |
| 144955144 | -1 | 56 | 0,0674647 | 144955143 |
| 150323853 | -1 | 56 | 0,0000073 | 150323852 |
| 155692562 | -1 | 56 | 0,0000053 | 155692561 |
| 161061271 | -1 | 56 | 0,0000056 | 161061270 |
| 166429980 | -1 | 56 | 0,0000124 | 166429979 |
| 171798689 | -1 | 56 | 0,0000057 | 171798688 |
| 177167398 | -1 | 56 | 0,0000058 | 177167397 |
| 182536107 | -1 | 56 | 0,0000054 | 182536106 |
| 187904816 | -1 | 56 | 0,0000095 | 187904815 |
| 193273525 | -1 | 56 | 0,0000056 | 193273524 |
| 198642234 | -1 | 56 | 0,0000057 | 198642233 |
| 204010943 | -1 | 56 | 0,0000137 | 204010942 |
| 209379652 | -1 | 56 | 0,000005 | 209379651 |
| 214748361 | -1 | 56 | 0,0000081 | 214748360 |
| 220117070 | -1 | 56 | 0,0000108 | 220117069 |

**Podsumowanie**

Algorytm wyszukiwania binarnego jest szybszy od wyszukiwania liniowego. Złożoność obliczeniowa wyszukiwania binarnego jest równa O(log) a wyszukiwania liniowego O (n). Pesymistyczna złożoność wyszukiwania liniowego jest linowa. Czas potrzebny na wyszukanie elementu metodą wyszukiwania liniowego jest dłuższy od czasu potrzebnego na metodę wyszukiwania liniowego.