ZADANIE

Projekt 1

do samodzielnego wykonania

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 2^{30} bajtów (2^{28} elementów typu *uint/int*).

W sprawozdaniu przedstawić dla każdego algorytmu:

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

I.Wyszukiwanie liniowe średnia złożoność.

Wyszukiwanie liniowe przed instrumentacją

```
static int wyszukiwanieliniowe(int[] tab, int num)//Wyszukiwanie przed instumentacją.
{
    for (int i = 0; i < tab.Length; i++)
    {
        if (tab[i] == num)
        {
            return i;
        }
    }
    return -1;
}
```

Wyszukiwanie liniowe z instrumentacją

```
private static int wyszukiwanieliniowepoinstumentacji(int[] tab, int num)//wyszukiwanie z instrumentacją.
{
    licznik = 0;
    suma = 0;
    for (int i = 0; i < tab.Length; i++)
    {
        ++licznik;
        suma += (long)tab[i];
        if (tab[i] == num)
        {
            return i;
        }
    }
    return -1;
}
```

Zaimplementowane wyszukiwanie w metodzie Main

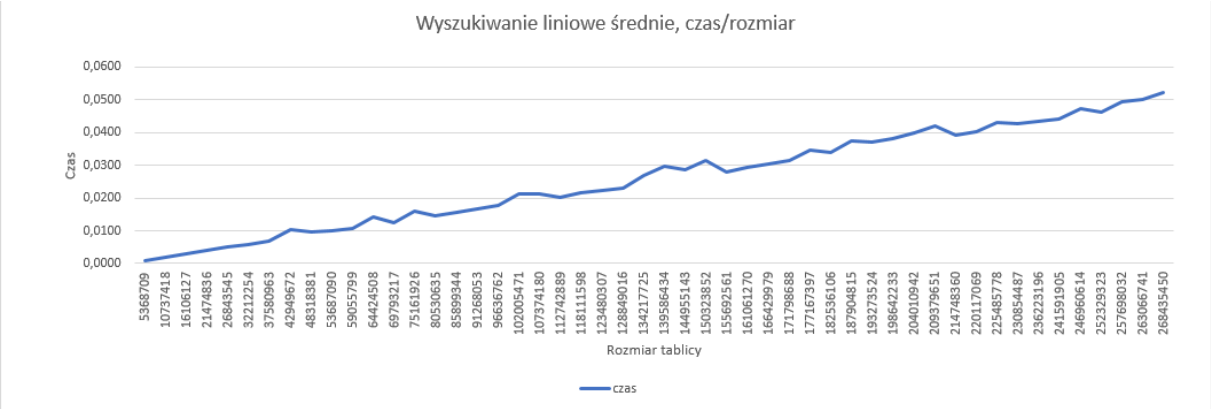
```
for (int i = 0; i < iter + 2; i++)
{
    szukana = table.Length / 2;
    long start = Stopwatch.GetTimestamp();//stoper start
    indeks = wyszukiwanieliniowe(table, szukana);//liniowe bez instrumentacji
    long stop = Stopwatch.GetTimestamp();//stoper stop
    iterTimeElapsed = stop - start;//za pomocą róznicy obliczamy czas
    TimeElapsed = iterTimeElapsed;
    indeks = wyszukiwanieliniowepoinstumentacji(table, szukana);//liniowe z instumentacją
}
```

Dane zgromadzone z 50 pomiarów

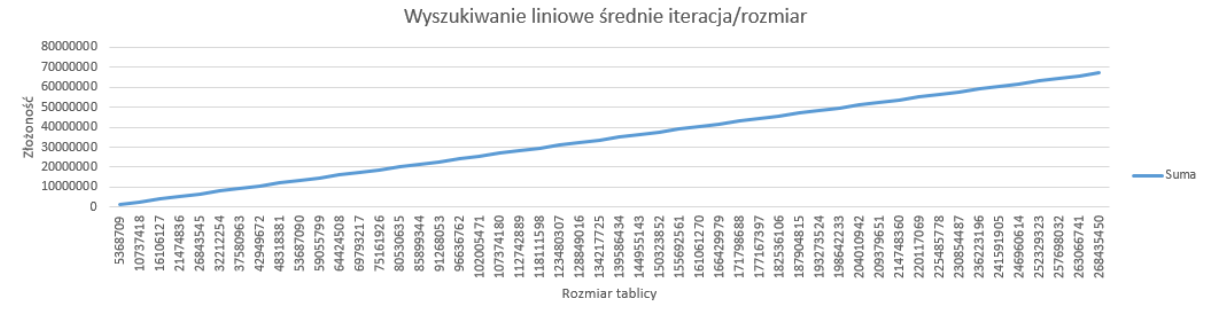
nr	rozmiar tablicy	szukana liczba	nr indeksu	liczba wykonanych operacji	czas	zlozonosc
1	5368709	2684354	2684353	2684354	0,0010	1342177
2	10737418	5368709	5368708	5368709	0,0021	2684355
3	16106127	8053063	8053062	8053063	0,0031	4026532
4	21474836	10737418	10737417	10737418	0,0039	5368709
5	26843545	13421772	13421771	13421772	0,0051	6710886
6	32212254	16106127	16106126	16106127	0,0059	8053064
7	37580963	18790481	18790480	18790481	0,0069	9395241
8	42949672	21474836	21474835	21474836	0,0102	10737418
9	48318381	24159190	24159189	24159190	0,0096	12079595
10	53687090	26843545	26843544	26843545	0,0100	13421773
11	59055799	29527899	29527898	29527899	0,0108	14763950
12	64424508	32212254	32212253	32212254	0,0142	16106127
13	69793217	34896608	34896607	34896608	0,0126	17448304
14	75161926	37580963	37580962	37580963	0,0160	18790482
15	80530635	40265317	40265316	40265317	0,0145	20132659
16	85899344	42949672	42949671	42949672	0,0155	21474836
17	91268053	45634026	45634025	45634026	0,0167	22817013
18	96636762	48318381	48318380	48318381	0,0177	24159191
19	102005471	51002735	51002734	51002735	0,0212	25501368
20	107374180	53687090	53687089	53687090	0,0211	26843545
21	112742889	56371444	56371443	56371444	0,0203	28185722
22	118111598	59055799	59055798	59055799	0,0215	29527900
23	123480307	61740153	61740152	61740153	0,0225	30870077
24	128849016	64424508	64424507	64424508	0,0232	32212254
25	134217725	67108862	67108861	67108862	0,0270	33554431
26	139586434	69793217	69793216	69793217	0,0296	34896609
27	144955143	72477571	72477570	72477571	0,0287	36238786
28	150323852	75161926	75161925	75161926	0,0315	37580963
29	155692561	77846280	77846279	77846280	0,0281	38923140
30	161061270	80530635	80530634	80530635	0,0294	40265318
31	166429979	83214989	83214988	83214989	0,0305	41607495
32	171798688	85899344	85899343	85899344	0,0314	42949672
33	177167397	88583698	88583697	88583698	0,0348	44291849

34	182536106	91268053	91268052	91268053	0,0341	45634027
35	187904815	93952407	93952406	93952407	0,0376	46976204
36	193273524	96636762	96636761	96636762	0,0371	48318381
37	198642233	99321116	99321115	99321116	0,0383	49660558
38	204010942	102005471	102005470	102005471	0,0400	51002736
39	209379651	104689825	104689824	104689825	0,0421	52344913
40	214748360	107374180	107374179	107374180	0,0391	53687090
41	220117069	110058534	110058533	110058534	0,0403	55029267
42	225485778	112742889	112742888	112742889	0,0430	56371445
43	230854487	115427243	115427242	115427243	0,0426	57713622
44	236223196	118111598	118111597	118111598	0,0433	59055799
45	241591905	120795952	120795951	120795952	0,0443	60397976
46	246960614	123480307	123480306	123480307	0,0474	61740154
47	252329323	126164661	126164660	126164661	0,0463	63082331
48	257698032	128849016	128849015	128849016	0,0494	64424508
49	263066741	131533370	131533369	131533370	0,0500	65766685
50	268435450	134217725	134217724	134217725	0,0521	67108863

Czas w porównaniu do wielkości tablicy



Iteracja w porównaniu do rozmiaru tablicy



II.Wyszukiwanie liniowe pesymistyczna złożoność.

Wyszukiwanie liniowe pesymistyczne przed instrumentacją

```
static int wyszukiwanieliniowe(int[] tab, int num)//bez instumentcji
{
    for (int i = 0; i < tab.Length; i++)
    {
        if (tab[i] == num)
        {
            return i;
        }
    }
    return -1;
}
```

Wyszukiwanie liniowe pesymistyczne z instrumentacją

```
private static int wyszukiwanieliniowepoinstumentacji(int[] tab, int num)//z instrumentacja
{
    licznik = 0;
    for (int i = 0; i < tab.Length; i++)
    {
        ++licznik;

        if (tab[i] == num)
        {
            return i;
        }
    }
    return -1;
}
```

Zaimplementowane wyszukiwanie w metodzie Main

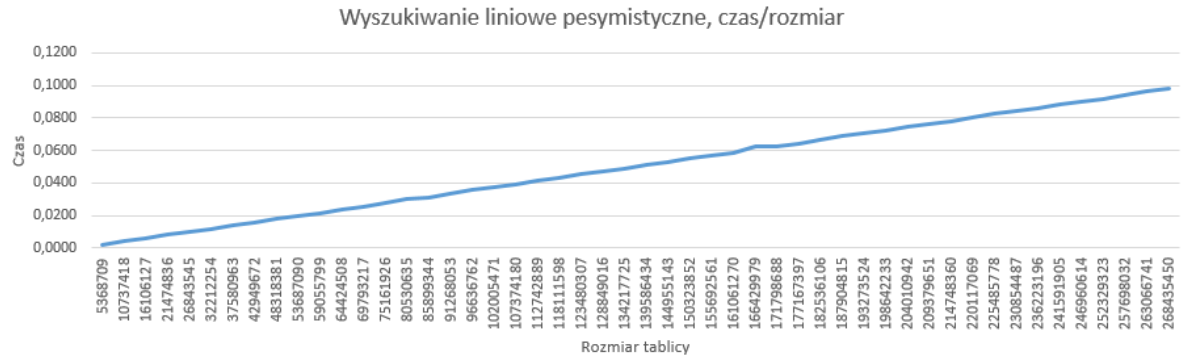
```
long start = Stopwatch.GetTimestamp();//stoper start
indeks = wyszukiwanieliniowe(table, szukana);//wyszukiwanie bez instrumentacji
long stop = Stopwatch.GetTimestamp();//stoper stop
iterTimeElapsed = stop - start;//czas obliczamy poprzez roznice
TimeElapsed = iterTimeElapsed;
indeks = wyszukiwanieliniowepoinstumentacji(table, szukana);//z instrumentacja
```

Dane zgromadzone z 50 pomiarów

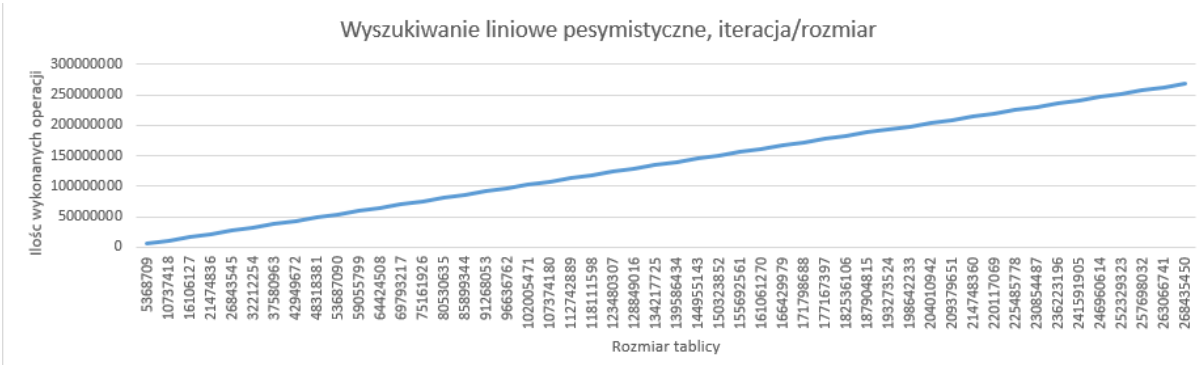
nr	rozmiar tablicy	szukana liczba	nr indeksu	ilość wykonanych operacji	czas
1	5368709	5368710	-1	5368709	0,0020
2	10737418	10737419	-1	10737418	0,0041
3	16106127	16106128	-1	16106127	0,0059
4	21474836	21474837	-1	21474836	0,0079
5	26843545	26843546	-1	26843545	0,0097
6	32212254	32212255	-1	32212254	0,0119
7	37580963	37580964	-1	37580963	0,0137
8	42949672	42949673	-1	42949672	0,0155
9	48318381	48318382	-1	48318381	0,0177
10	53687090	53687091	-1	53687090	0,0195

11	59055799	59055800	-1	59055799	0,0216
12	64424508	64424509	-1	64424508	0,0234
13	69793217	69793218	-1	69793217	0,0254
14	75161926	75161927	-1	75161926	0,0276
15	80530635	80530636	-1	80530635	0,0303
16	85899344	85899345	-1	85899344	0,0313
17	91268053	91268054	-1	91268053	0,0334
18	96636762	96636763	-1	96636762	0,0354
19	102005471	102005472	-1	102005471	0,0372
20	107374180	107374181	-1	107374180	0,0392
21	112742889	112742890	-1	112742889	0,0411
22	118111598	118111599	-1	118111598	0,0432
23	123480307	123480308	-1	123480307	0,0454
24	128849016	128849017	-1	128849016	0,0473
25	134217725	134217726	-1	134217725	0,0490
26	139586434	139586435	-1	139586434	0,0509
27	144955143	144955144	-1	144955143	0,0528
28	150323852	150323853	-1	150323852	0,0553
29	155692561	155692562	-1	155692561	0,0569
30	161061270	161061271	-1	161061270	0,0588
31	166429979	166429980	-1	166429979	0,0625
32	171798688	171798689	-1	171798688	0,0627
33	177167397	177167398	-1	177167397	0,0645
34	182536106	182536107	-1	182536106	0,0664
35	187904815	187904816	-1	187904815	0,0689
36	193273524	193273525	-1	193273524	0,0704
37	198642233	198642234	-1	198642233	0,0724
38	204010942	204010943	-1	204010942	0,0744
39	209379651	209379652	-1	209379651	0,0764
40	214748360	214748361	-1	214748360	0,0781
41	220117069	220117070	-1	220117069	0,0807
42	225485778	225485779	-1	225485778	0,0826
43	230854487	230854488	-1	230854487	0,0845
44	236223196	236223197	-1	236223196	0,0860
45	241591905	241591906	-1	241591905	0,0880
46	246960614	246960615	-1	246960614	0,0901
47	252329323	252329324	-1	252329323	0,0916
48	257698032	257698033	-1	257698032	0,0939
49	263066741	263066742	-1	263066741	0,0967
50	268435450	268435451	-1	268435450	0,0981

Czas w porównaniu do rozmiaru tablicy



Iteracja w porównaniu do rozmiaru tablicy



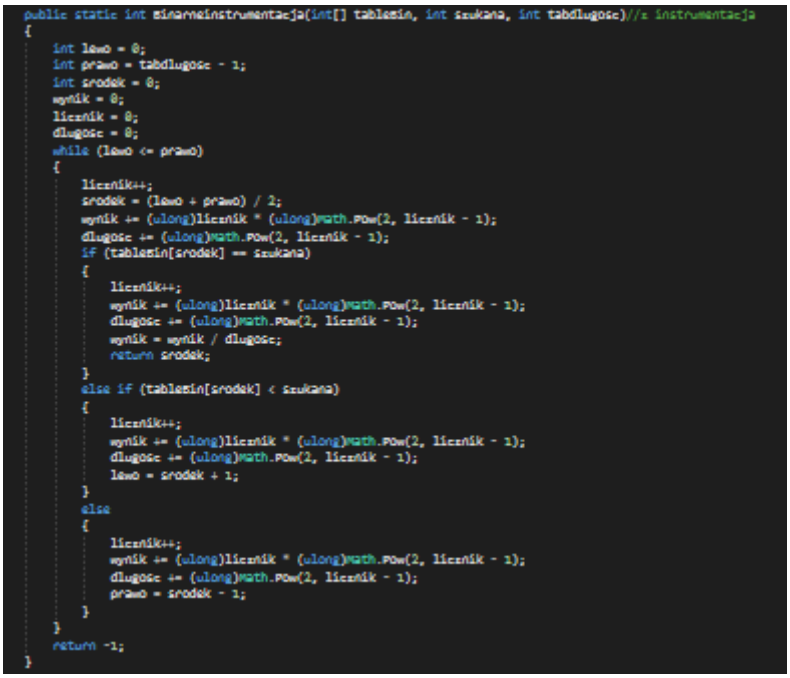
III.Wyszukiwanie binarne średnia złożoność.

Wyszukiwanie binarne przed instrumentacją.

```
public static int Binarne(int[] tableBin, int szukana, int tabdlugosc)//bez instrumentacji
{
    int lewo = 0;
    int prawo = tabdlugosc - 1;
    int srodek = 0;
    while (lewo <= prawo)
    {
        srodek = (lewo + prawo) / 2;
        if (tableBin[srodek] == szukana)
        {
            return srodek;
        }
        else if (tableBin[srodek] < szukana)
        {
            lewo = srodek + 1;
        }
        else
        {

            prawo = srodek - 1;
        }
    }
    return -1;
}
```

Wyszukiwanie binarne z instrumentacją.



```
public static int Binarneinstrumentacja(int[] tablicin, int szukana, int tabdlugosc)//z instrumentacja
{
    int lewo = 0;
    int prawo = tabdlugosc - 1;
    int srodek = 0;
    wynik = 0;
    licznik = 0;
    dlugosc = 0;
    while (lewo <= prawo)
    {
        licznik++;
        srodek = (lewo + prawo) / 2;
        wynik += (ulong)licznik * (ulong)Math.Pow(2, licznik - 1);
        dlugosc += (ulong)Math.Pow(2, licznik - 1);
        if (tablicin[srodek] == szukana)
        {
            licznik++;
            wynik += (ulong)licznik * (ulong)Math.Pow(2, licznik - 1);
            dlugosc += (ulong)Math.Pow(2, licznik - 1);
            wynik = wynik / dlugosc;
            return srodek;
        }
        else if (tablicin[srodek] < szukana)
        {
            licznik++;
            wynik += (ulong)licznik * (ulong)Math.Pow(2, licznik - 1);
            dlugosc += (ulong)Math.Pow(2, licznik - 1);
            lewo = srodek + 1;
        }
        else
        {
            licznik++;
            wynik += (ulong)licznik * (ulong)Math.Pow(2, licznik - 1);
            dlugosc += (ulong)Math.Pow(2, licznik - 1);
            prawo = srodek - 1;
        }
    }
    return -1;
}
```

Wyszukiwanie zaimplementowane w metodzie Main

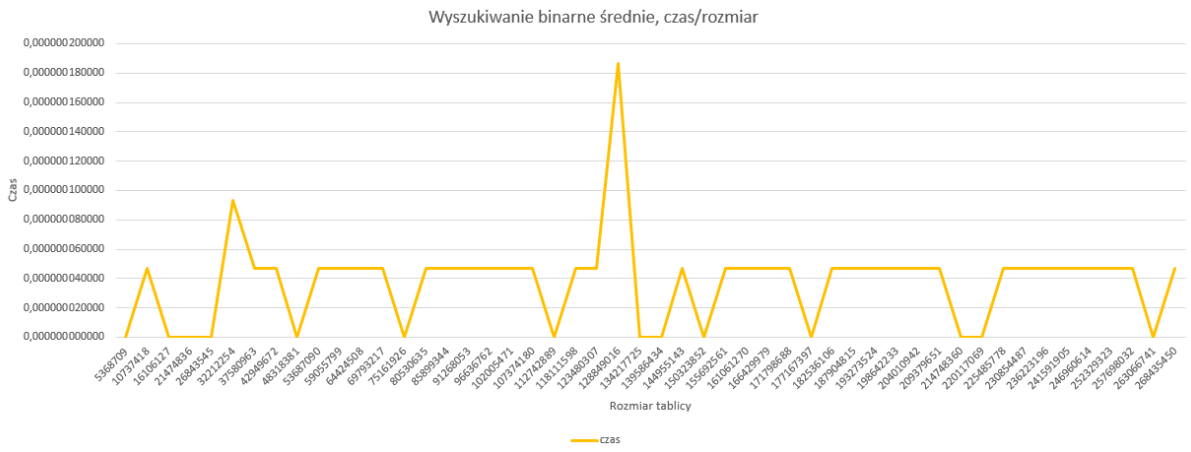
```
long start = Stopwatch.GetTimestamp();//stoper start
indeks = Binarne(table, szukana, table.Length);//bez instrumentacji
long stop = Stopwatch.GetTimestamp();//stoper stop
iterTimeElapsed = stop - start;//czas obliczamy za pomoca roznicy
TimeElapsed = iterTimeElapsed;
indeks = Binarneinstrumentacja(table, szukana, table.Length);//z instrumentacja
```

Dane zgromadzone z 50 pomiarów

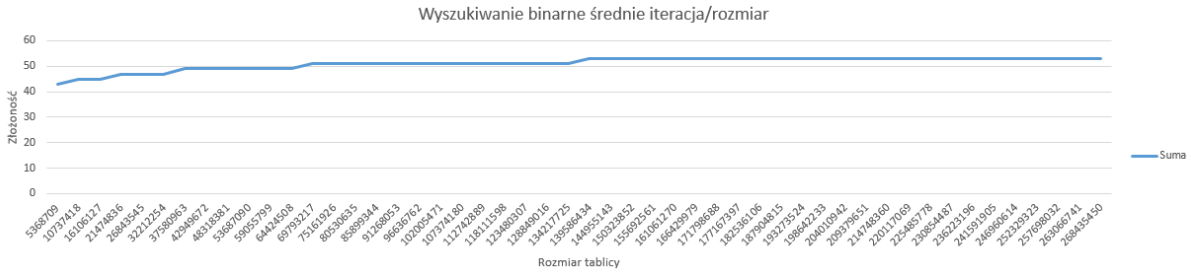
nr	rozmiar tablicy	szukana	indeks	ilość operacji	czas	złożoność
1	5368709	5368708	5368707	44	0,000000000000	43
2	10737418	10737417	10737416	46	0,000000046613	45
3	16106127	16106126	16106125	46	0,000000000000	45
4	21474836	21474835	21474834	48	0,000000000000	47
5	26843545	26843544	26843543	48	0,000000000000	47
6	32212254	32212253	32212252	48	0,000000093226	47
7	37580963	37580962	37580961	50	0,000000046613	49
8	42949672	42949671	42949670	50	0,000000046613	49
9	48318381	48318380	48318379	50	0,000000000000	49
10	53687090	53687089	53687088	50	0,000000046613	49
11	59055799	59055798	59055797	50	0,000000046613	49
12	64424508	64424507	64424506	50	0,000000046613	49
13	69793217	69793216	69793215	52	0,000000046613	51
14	75161926	75161925	75161924	52	0,000000000000	51
15	80530635	80530634	80530633	52	0,000000046613	51
16	85899344	85899343	85899342	52	0,000000046613	51
17	91268053	91268052	91268051	52	0,000000046613	51

18	96636762	96636761	96636760	52	0,000000046613	51
19	102005471	102005470	102005469	52	0,000000046613	51
20	107374180	107374179	107374178	52	0,000000046613	51
21	112742889	112742888	112742887	52	0,000000000000	51
22	118111598	118111597	118111596	52	0,000000046613	51
23	123480307	123480306	123480305	52	0,000000046613	51
24	128849016	128849015	128849014	52	0,000000186451	51
25	134217725	134217724	134217723	52	0,000000000000	51
26	139586434	139586433	139586432	54	0,000000000000	53
27	144955143	144955142	144955141	54	0,000000046613	53
28	150323852	150323851	150323850	54	0,000000000000	53
29	155692561	155692560	155692559	54	0,000000046613	53
30	161061270	161061269	161061268	54	0,000000046613	53
31	166429979	166429978	166429977	54	0,000000046613	53
32	171798688	171798687	171798686	54	0,000000046613	53
33	177167397	177167396	177167395	54	0,000000000000	53
34	182536106	182536105	182536104	54	0,000000046613	53
35	187904815	187904814	187904813	54	0,000000046613	53
36	193273524	193273523	193273522	54	0,000000046613	53
37	198642233	198642232	198642231	54	0,000000046613	53
38	204010942	204010941	204010940	54	0,000000046613	53
39	209379651	209379650	209379649	54	0,000000046613	53
40	214748360	214748359	214748358	54	0,000000000000	53
41	220117069	220117068	220117067	54	0,000000000000	53
42	225485778	225485777	225485776	54	0,000000046613	53
43	230854487	230854486	230854485	54	0,000000046613	53
44	236223196	236223195	236223194	54	0,000000046613	53
45	241591905	241591904	241591903	54	0,000000046613	53
46	246960614	246960613	246960612	54	0,000000046613	53
47	252329323	252329322	252329321	54	0,000000046613	53
48	257698032	257698031	257698030	54	0,000000046613	53
49	263066741	263066740	263066739	54	0,000000000000	53
50	268435450	268435449	268435448	54	0,000000046613	53

Czas w porównaniu do rozmiaru tablicy



Iteracja w porównaniu do rozmiaru tablicy



IV.Wyszukiwanie binarne pesymistyczna złożoność.

Wyszukiwanie przed instrumentacją

```
public static int Binarne(int[] tabBin, int szukanaliczba, int tabdlugosc)//wyszukiwanie bez instrumentacji
{
    int lewa = 0;
    int prawa = tabdlugosc - 1;
    int srodek = 0;
    while (lewa <= prawa)
    {
        srodek = (lewa + prawa) / 2;
        if (tabBin[srodek] == szukanaliczba)
        {
            return srodek;
        }
        else if (tabBin[srodek] < szukanaliczba)
        {
            lewa = srodek + 1;
        }
        else
        {
            prawa = srodek - 1;
        }
    }
    return -1;
}
```

Wyszukiwanie z instrumentacją

```
public static int Binarneinstrumentacja(int[] tableBin, int szukanaliczba, int tabdlugosc)//z instrumentacja
{
    int lewa = 0;
    int prawa = tabdlugosc - 1;
    int srodek = 0;
    licznik = 0;
    while (lewa <= prawa)
    {
        licznik++;
        srodek = (lewa + prawa) / 2;
        if (tableBin[srodek] == szukanaliczba)
        {
            licznik++;
            return srodek;
        }
        else if (tableBin[srodek] < szukanaliczba)
        {
            licznik++;
            lewa = srodek + 1;
        }
        else
        {
            licznik++;
            prawa = srodek - 1;
        }
    }
    return -1;
}
```

Wyszukiwanie zaimplementowane w metodzie Main

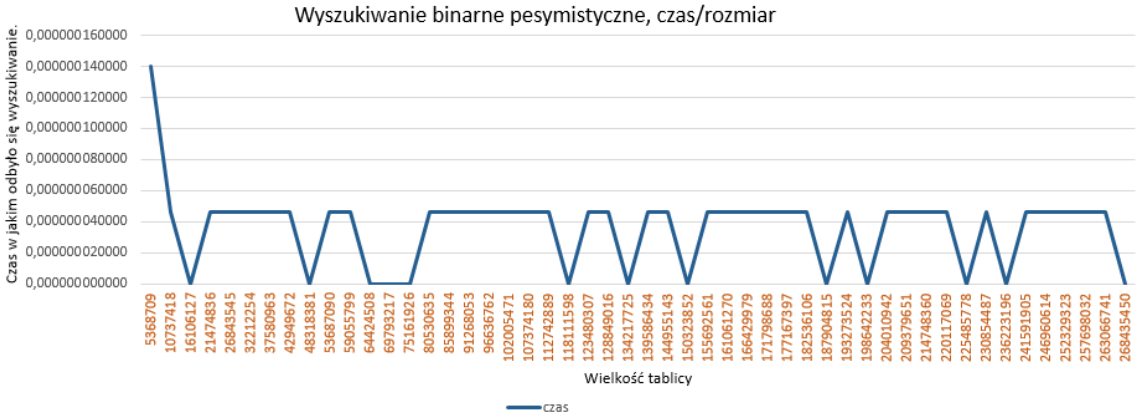
```
long start = Stopwatch.GetTimestamp();//stoper start
indeks = Binarne(table, szukana, table.Length);//wyszukiwane z instrumentacja
long stop = Stopwatch.GetTimestamp();//stoper stop
iterTimeElapsed = stop - start;//czas obliczamy za pomoca roznicy
TimeElapsed = iterTimeElapsed;
indeks = Binarneinstrumentacja(table, szukana, table.Length);//wyszukiwane z instrumentacja
```

Dane zgromadzone z 50 pomiarów

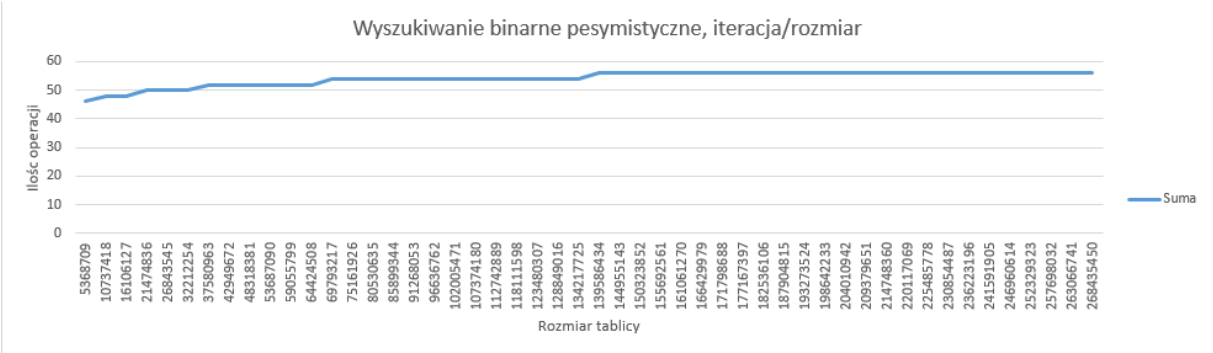
nr	rozmiar tablicy	szukana liczba	indeks	ilość wykonanych operacji	czas
1	5368709	5368710	-1	46	0,000000139839
2	10737418	10737419	-1	48	0,000000046613
3	16106127	16106128	-1	48	0,000000000000
4	21474836	21474837	-1	50	0,000000046613
5	26843545	26843546	-1	50	0,000000046613
6	32212254	32212255	-1	50	0,000000046613
7	37580963	37580964	-1	52	0,000000046613
8	42949672	42949673	-1	52	0,000000046613
9	48318381	48318382	-1	52	0,000000000000
10	53687090	53687091	-1	52	0,000000046613
11	59055799	59055800	-1	52	0,000000046613
12	64424508	64424509	-1	52	0,000000000000
13	69793217	69793218	-1	54	0,000000000000
14	75161926	75161927	-1	54	0,000000000000
15	80530635	80530636	-1	54	0,000000046613
16	85899344	85899345	-1	54	0,000000046613
17	91268053	91268054	-1	54	0,000000046613
18	96636762	96636763	-1	54	0,000000046613
19	102005471	102005472	-1	54	0,000000046613
20	107374180	107374181	-1	54	0,000000046613
21	112742889	112742890	-1	54	0,000000046613
22	118111598	118111599	-1	54	0,000000000000
23	123480307	123480308	-1	54	0,000000046613
24	128849016	128849017	-1	54	0,000000046613
25	134217725	134217726	-1	54	0,000000000000
26	139586434	139586435	-1	56	0,000000046613
27	144955143	144955144	-1	56	0,000000046613
28	150323852	150323853	-1	56	0,000000000000
29	155692561	155692562	-1	56	0,000000046613
30	161061270	161061271	-1	56	0,000000046613
31	166429979	166429980	-1	56	0,000000046613
32	171798688	171798689	-1	56	0,000000046613
33	177167397	177167398	-1	56	0,000000046613
34	182536106	182536107	-1	56	0,000000046613
35	187904815	187904816	-1	56	0,000000000000
36	193273524	193273525	-1	56	0,000000046613

37	198642233	198642234	-1	56	0,000000000000
38	204010942	204010943	-1	56	0,000000046613
39	209379651	209379652	-1	56	0,000000046613
40	214748360	214748361	-1	56	0,000000046613
41	220117069	220117070	-1	56	0,000000046613
42	225485778	225485779	-1	56	0,000000000000
43	230854487	230854488	-1	56	0,000000046613
44	236223196	236223197	-1	56	0,000000000000
45	241591905	241591906	-1	56	0,000000046613
46	246960614	246960615	-1	56	0,000000046613
47	252329323	252329324	-1	56	0,000000046613
48	257698032	257698033	-1	56	0,000000046613
49	263066741	263066742	-1	56	0,000000046613
50	268435450	268435451	-1	56	0,000000000000

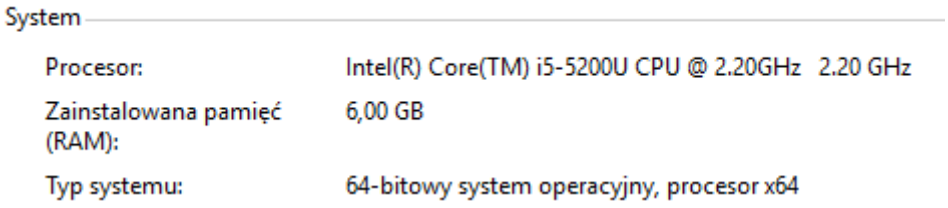
Czas w porównaniu do wielkości tablicy



Iteracja w porównaniu do rozmiaru tablicy



Komputer, na którym zostały wykonane pomiary



Wnioski

- Wyszukiwanie binarne jest dużo szybsze co widać na zamieszczonych wyżej danych.
- Czas wyszukiwania liniowego w złożoności średniej i pesymistycznej widocznie się różni.
- Czasy wyszukiwania binarnego w złożoności średniej i pesymistycznej są podobne.