

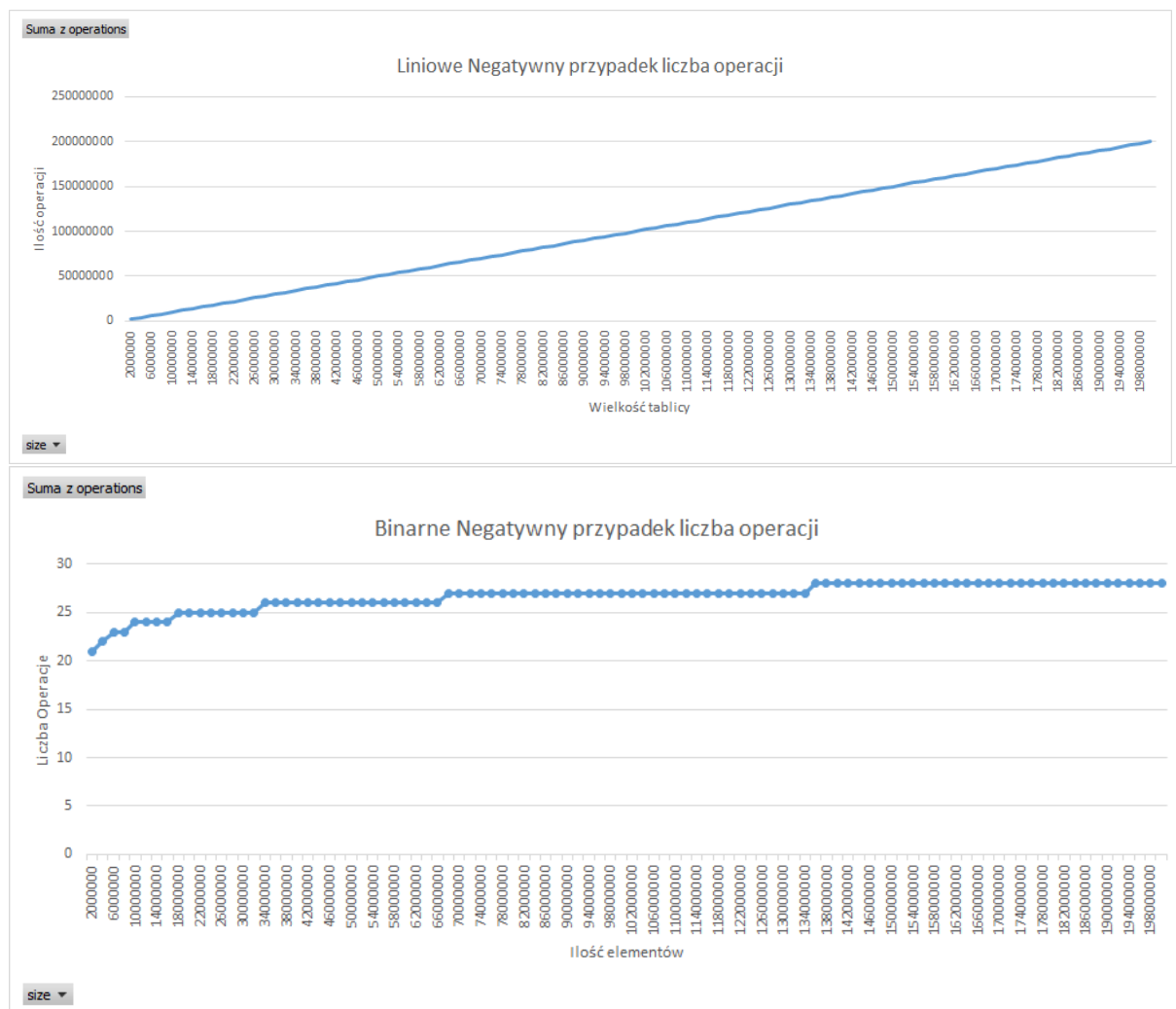
Projekt 1

W tym projekcie zajmę się badaniem efektywności wyszukiwania liniowego i binarnego pod względem ilości operacji oraz czasie potrzebnym na wykonanie. Operacje będą przeprowadzać na tablicy której liczba elementów będzie wzrastać z początkowej wartości 2 000 000 do 200 000 000 inkrementując się o 2 000 000. Tablice zostaną wypełnione losowymi liczbami z przedziału od 1 do 1000. Dodatkowo dokonam pomiarów na tablicy 2^{28} elementów. Rozważę przypadek negatywny oraz przypadek średni. Program został napisany w języku C#.

Czas każdorazowo był mierzony 12 razy, a następnie uśredniany. Miało to na celu uzyskanie bardziej miarodajnych wyników.

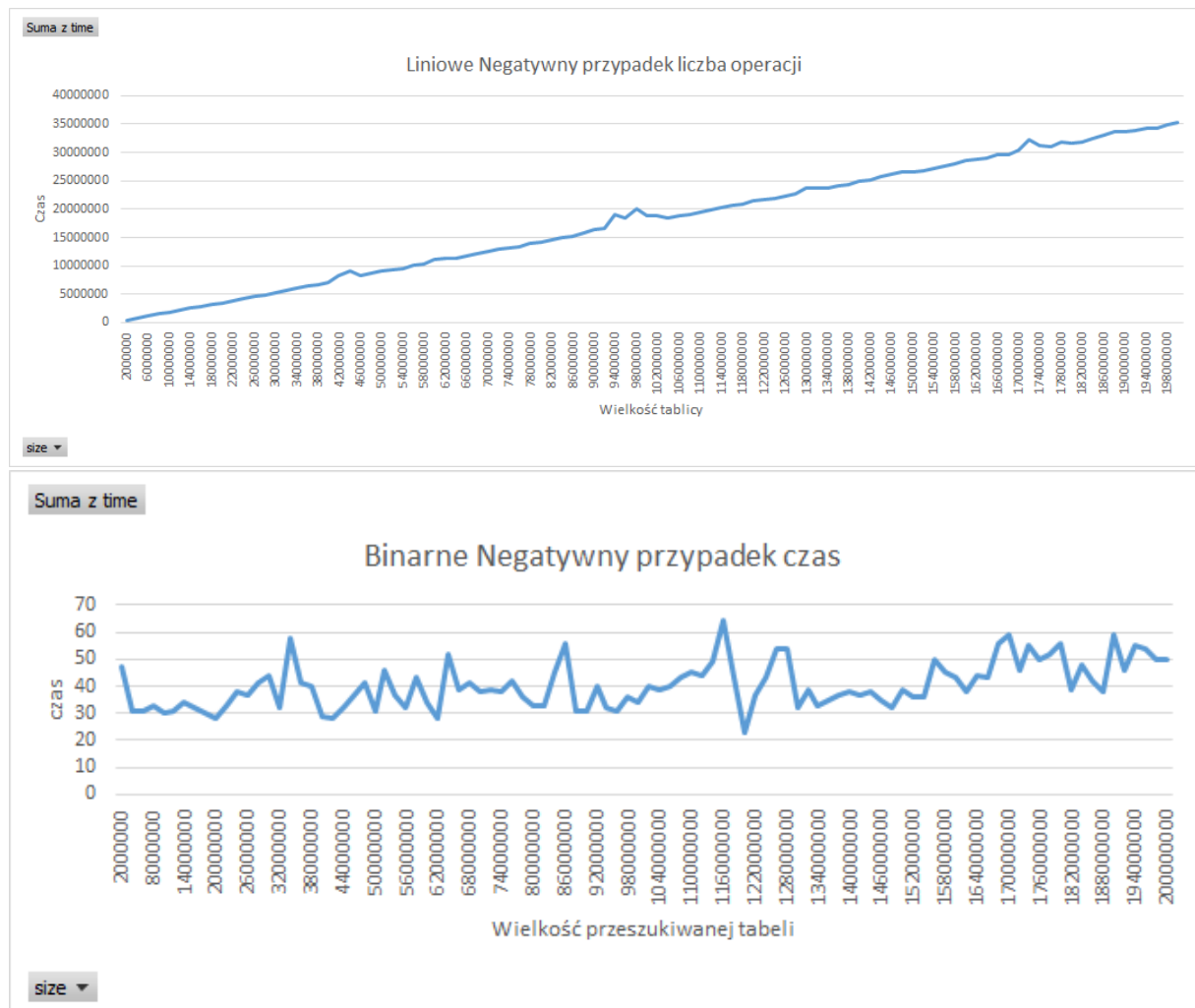
Pomiary zostały wykonane na jednostce operującej na 3GB pamięci podręcznej, oraz na procesorze: Intel Xeon X5450 3.00GHz.

Przypadek negatywny ilość operacji



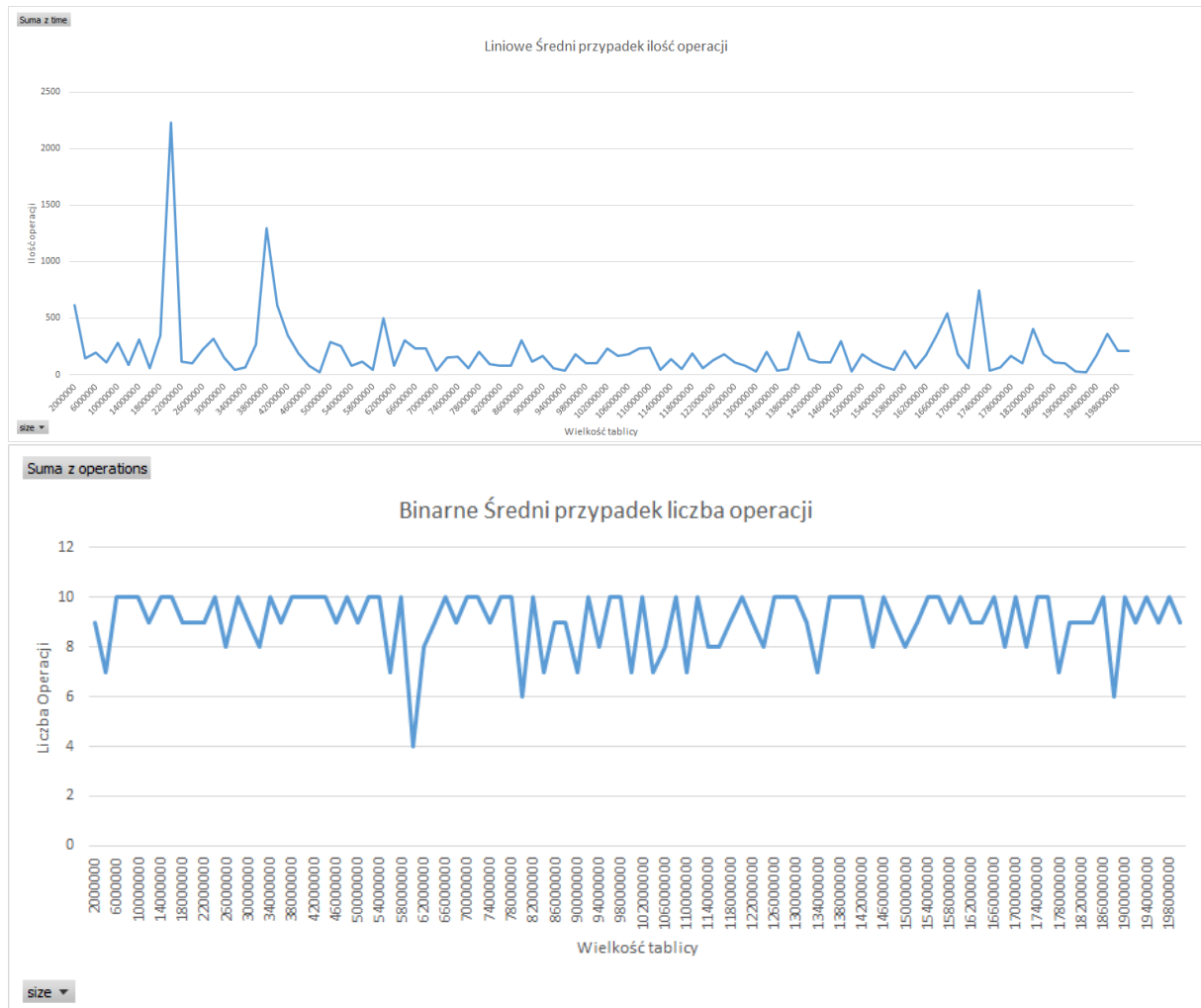
Na podstawie powyższych wykresów, możemy jasno zauważyć, że wykres wyszukiwania liniowego przyjmuje obraz funkcji liniowej, natomiast wyszukiwanie binarne przyjęło wykres funkcji logarytmicznej. Podczas wyszukiwania liniowego ilość operacji jest wprost proporcjonalna do wielkości tablicy, natomiast wyszukiwanie binarne osiąga znacząco mniejszą ilość operacji nawet w przypadku tablicy zawierającej 400 000 000 elementów

Przypadek negatywny czas



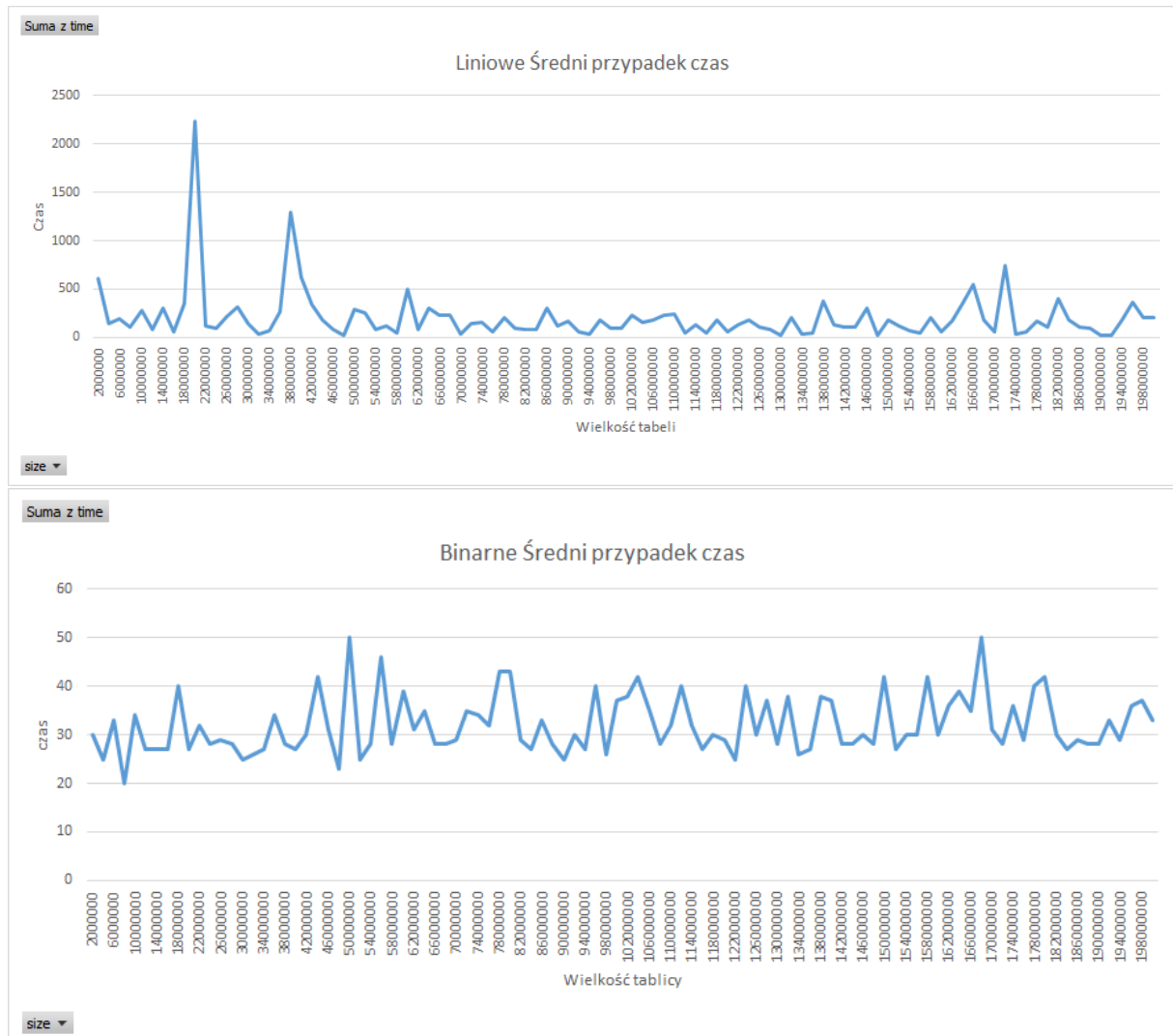
Analizując czas potrzebny na wykonanie pętli, możemy zauważyć, że znowu wyszukiwanie liniowe przyjmuje wykres przybliżony to funkcji liniowej. Posiada on pewne odchylenia, aczkolwiek uważam, że powstały one w skutek spowolnienia pracy procesora kiedy skupiał się na zarządzaniu dostępną pamięcią. W porównaniu do funkcji wyszukiwania liniowego, wyszukiwanie binarne przyjmuje praktycznie obraz funkcji stałej. Nie jest to widoczne na załączonym wykresie z powodu skali. Wyszukiwanie binarne radziło sobie z rozwiązaniem pętli w przedziale 20ms do 65ms, natomiast w przypadku wyszukiwania liniowego czas ten był znacząco większy.

Przypadek średni ilość operacji



Badając przypadek średni wyszukiwałem liczby z zakresu od 1 do 1000. W tym eksperymencie dużą rolę pełnił czynnik losowy. Jednakże najważniejszy wniosek narzuca się na pierwszy rzut oka. Największa ilość operacji w przypadku wyszukiwania binarnego to 10. Jest to znacząca różnica względem wyszukiwania liniowego którego średnia ilość operacji wynosiła między 100 a 500, z uwagi na losowość zdarzały się przypadki gdzie wymagana liczba operacji przekraczała 1000.

Przypadek średni czas



Po analizie ilości operacji, wykresy czasu prezentują się praktycznie identycznie. Można zatem wysunąć wniosek, że ilość operacji ma bezpośredni wpływ na czas potrzebny podczas wyszukiwania. Ponownie wyszukiwanie binarne zwycięża posiadając niezwykle niski czas potrzebny na realizację.

Tablica 2^{28} (268 435 456) elementów

| | Ilość Operacji | Czas |
|----------------------|----------------|----------|
| Przypadek Negatywny | | |
| Wyszukiwanie Liniowe | 268435456 | 48026432 |
| Wyszukiwanie Binarne | 29 | 802 |
| Przypadek Średni | | |
| Wyszukiwanie Liniowe | 7317 | 974 |
| Wyszukiwanie Binarne | 10 | 17468 |

W przypadku tablicy o 2^{28} elementów również pomiar ilości operacji został wykonany 12 razy a następnie uśredniony w celu bardziej dokładnych danych.

Na przykładzie tego eksperymentu możemy znowu zauważyć przewagę wyszukiwania binarnego nad wyszukiwaniem liniowym, jeżeli chodzi o ilość operacji i czas. Niestety dostrzegam błąd który powstaje w przypadku pomiaru czasu przypadku średniego za pomocą wyszukiwania binarnego, ponieważ potrzebny czas na wykonanie pętli jest dłuższy niż w przypadku negatywnym. Jednakże pomimo tego błędu możemy zauważyć ogromną przewagę wyszukiwania binarnego w średniej ilości potrzebnych operacji nad wyszukiwaniem liniowym.

Podsumowanie

Niestety podczas analizy pojawiły się pewne błędy pomiarowe, które wynikają prawdopodobnie ze zbyt małej pamięci operacyjnej. Jednakże błędy te nie mogą podważyć werdyktów iż wyszukiwanie binarne jest o wiele wydajniejsze pod względem czasu i liczby operacji nad wyszukiwaniem liniowym. Jedynym minusem wyszukiwania binarnego jest to, że uprzednio należy tablicę posortować. Mimo tego uważam, że pomimo operacji sortowania i tak lepiej jest używać wyszukiwania binarnego.